N° d'ordre : 192/ 2019

UNIVERSITE SULTAN MOULAY SLIMANE

Faculté des Sciences et Techniques

Béni-Mellal

Centre d'Etude Doctorales : **Sciences et Techniques**

Formation doctorale : **Mathématiques et Physique Appliquées**

THÈSE

Présentée par

Abdelkarim BEN-CHARKE

Pour l'obtention du grade de

DOCTEUR

Spécialité **: Informatique**

Option **: Informatique**

Vers la sécurité du système d'information : Application des IDS (Snort et Suricata)

Soutenue le 16/02/2019 à 09h30 devant la commission d'examen composée de:

Brahim MINAOUI	:	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni Mellal, Maroc — Président
Abdellatif HAIR	:	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni Mellal, Maroc — Rapporteur
Rachid EL AYACHI	:	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni Mellal, Maroc — Rapporteur
Jilali ANTARI	:	Professeur, Université Ibnou Zohr, Faculté Polydisciplinaire, Taroudant, Maroc — Rapporteur
Mohamed BASLAM	:	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni Mellal, Maroc — Examinateur
Mohamed CHABI	:	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni Mellal, Maroc — Encadrant
Mohamed FAKIR	:	Professeur, Université Sultan Moulay Slimane, F.S.T. Béni Mellal, Maroc — Co-encadrant

Remerciement

Je tiens à remercier très sincèrement toutes les personnes qui, par leurs conseils et leurs encouragements ont contribué à l'aboutissement de ma thèse de doctorat.

Mes remerciements sont adressés tout d'abord à mon directeur de thèse le professeur Mohamed Chabi et mon co-encadrant le professeur Mohamed Fakir pour avoir accepté de m'encadrer et pour leurs soutien constant pendant la durée de la thèse. Ils furent pour moi des encadrants attentifs et disponibles malgré leurs charges nombreuses. Leurs compétences, leurs clairvoyances, leurs charismes et leur dynamisme m'ont beaucoup appris. Ils ont été et resteront des moteurs dans mon travail.

Je voudrais exprimer ma gratitude à Monsieur Abdellatif HAIR, Professeur à la Faculté des Sciences et Techniques de Béni Mellal, à Monsieur Rachid EL AYACHI, Professeur à la Faculté des Sciences et Techniques de Béni Mellal, et à Monsieur Jilali ANTARI, Professeur à la Faculté Polydisciplinaire de Taroudant d'avoir accepté la charge d'être rapporteurs de ma thèse.

Je suis reconnaissant à Monsieur Mohamed BASLAM, Professeur à la Faculté des Sciences et Techniques de Béni Mellal pour l'honneur d'être examinateur de ma thèse.

Je remercie enfin Monsieur Brahim MINAOUI, Professeur à la Faculté des Sciences et Techniques de Béni Mellal pour m'avoir fait l'honneur d'accepter de présider le jury de cette thèse.

Un merci tout particulier à mes amis et mes collègues. Leur amitié, leur confiance en moi et leur soutien généreux m'ont été indispensables. Je garde pour eux et pour toujours, une place particulière dans mon cœur.

Les mots les plus simples étant les plus forts, j'adresse toute mon affection à ma petite famille, et en particulier à ma femme, à mes filles. Leur confiance, leur tendresse, leur amour me portent et me guident tous les jours.

Enfin, je réserve une reconnaissance particulière à mes chers parents, pour l'amour et le soutien incomparable dont ils m'ont fait preuve depuis ma naissance.

Résumé

Le système d'information prend une place stratégique au sein d'une entreprise où dans une organisation. Il permet aux différents acteurs de véhiculer leurs informations et de se communiquer grâce à un ensemble de ressources matérielles, humaines et logicielles. Dans cette thèse nous avons proposé une solution de la sécurité et de la protection d'accès aux systèmes informatiques traditionnelles aussi que nous avons apporté des solutions de contrôle d'accès et de surveillance pour les systèmes qui utilisent les infrastructures virtuelles. Pour le système traditionnel, nous avons travaillé sur la sécurité du système d'information par l'exploit des techniques de la cryptographie, les protocoles de sécurité comme SSH, SSL, IPSEC, la mise en place des Firewall pour le filtrage des paquets échangés et pour contrôler des accès. Nous avons exploité aussi la mise en place d'un système de détection d'intrusion (SNORT) afin de surveiller un système d'information. La virtualisation permet d'optimiser la charge de travail des serveurs physiques, de réduire l'infrastructure physique et d'économiser de l'énergie, offre la reprise automatique lors d'un incident, et aussi facilite la gestion de l'information. Partant, le problème de sécurité de l'information est une préoccupation sérieuse de l'entreprise. Divers travaux de recherche ont traité le problème de la sécurité de l'information, mais malgré ça, ce domaine souffre encore des intrusions. Face à ce problème de sécurité, nous avons proposé une solution de contrôle d'accès aux infrastructures virtuelles open sources par l'exploitation d'un système de prévention d'intrusion open source Suricata.

Mots clés : Système d'information, Sécurité des SI, Chiffrement, Déchiffrement, Firewall, Attaques réseau, Système de détection d'intrusion, Virtualisation, ELK, Suricata, Snort, Évaluation des Performances

Table de matières

Liste des figures

Liste des tables

Acronymes

AH	:	Authentication Header
DAS	:	Direct Attached Storage
ELK	:	Elasticksearch LogstashKibana
ESP	:	Encapsulating Security Payload
FN	:	False Negative
FNR	:	False Negative Rate
FPR	:	False Positive Rate
HIDS	:	Host-based Intrusion Detection System
HMAC	:	Hash Message Authentication code
IDS	:	Intrusion Detection System
IKE	:	Internet Key Exchange
ISAKM	:	Internet Security Association and Key Management Protocol
KVM	:	Kernel-based Virtual Machine
MAC	:	Message Authentication Code
MD5	:	Message Digest 5
NAS	:	Networked Attached Storage
NIDS	:	Network based Intrusion Detection System
SA	:	Security association
SAN	:	Storage Area Network
SI	:	Systèmed'Information
SSH	:	Secure Shell
SSL	:	Secure Socket Layer
TP	:	True Positive
TPR	:	True Positive Rate
VLAN	:	Virtual Local Area Network
VM	:	Virtual Machine

Introduction générale

La numérisation de l'entreprise passe par la création d'un système d'information(SI). Ce système est composé de ressources nécessaires telles que le personnel, le matériel, les logiciels ainsi que les procédures et les fonctions capables de collecter, gérer et stocker les informations et les données en vues d'une restitution ultérieure qui soit le plus fiable. Ce que rend ce système occupe une place stratégique au sein d'une entreprise.

L'utilisation des systèmes informatiques est aujourd'hui en pleine évolution. Le modèle classique qui consiste à associer à chaque utilisateur une machine physique qu'il possède et dont il va exploiter les ressources, devient de plus en plus obsolète. De nos jours, les ressources informatiques que l'on utilise peuvent être distribuées n'importe où dans l'Internet et les postes de travail du quotidien ne sont plus systématiquement des machines réelles. Cette constatation met en avant deux phénomènes importantes qui sont à l'origine de l'évolution de notre utilisation de l'informatique : l'informatique en nuage (Cloud computing) et la virtualisation. L'informatique en nuage, est l'exploitation de la puissance de calcul ou de stockage de serveurs informatiques distants par l'intermédiaire d'un réseau, généralement Internet. Ces serveurs sont loués à la demande, selon des critères techniques, mais, également, au forfait. l'informatique en nuage se caractérise par sa grande souplesse : selon le niveau de compétence de l'utilisateur client, il est possible de gérer soi-même son serveur ou de se contenter d'utiliser des applicatifs distants. Selon la définition du National Institute of Standards and Technology (NIST), l'informatique en nuage est l'accès via un réseau de télécommunications, à la demande et en libre-service, à des ressources informatiques partagées configurables. Il s'agit donc d'une délocalisation de l'infrastructure informatique. Une manière pour atteindre ces objectifs nécessite notamment l'utilisation de machines virtuelles et des techniques de virtualisation associées. Même si la virtualisation de ressources informatiques n'est pas née avec le Cloud, l'avènement du Cloud a considérablement augmenté son utilisation. L'ensemble des fournisseurs d'informatique en nuage s'appuient actuellement sur des machines virtuelles, qui sont beaucoup plus facilement déployables et migrables que des machines réelles. La virtualisation de ressources informatiques était auparavant essentiellement basée sur des techniques logicielles. Mais l'utilisation massive de machines virtuelles notamment pour le Cloud, a poussé les fondeurs de processeurs à inclure des mécanismes d'assistance matérielle à la virtualisation dans leurs processeurs. Ces extensions matérielles permettent d'une part de rendre plus facile la virtualisation, et d'autre part d'obtenir des gains de performance. Ainsi, un certain nombre de technologies ont vu le jour, telles que VT-x et VT-d chez Intel ou AMDV chez AMD ou Virtualization Extensions chez ARM. Par ailleurs, la virtualisation nécessite l'implémentation de fonctionnalités supplémentaires, capables de gérer les différentes machines virtuelles, de pouvoir les ordonnancer, les isoler et partager les ressources matérielles comme la mémoire et les périphériques. Ces différentes fonctionnalités sont en général prises en charge par un gestionnaire de machines virtuelles, dont le travail peut donc être plus ou moins fluide, en fonction des caractéristiques du processeur sur lequel il s'exécute. De façon globale, ces technologies introduisent des nouveaux modes d'exécution sur les processeurs, de plus en plus privilégiés et de plus en plus complexes. Ainsi, s'il est indéniable que l'utilisation de la virtualisation apporte un véritable intérêt pour l'informatique d'aujourd'hui, il est par ailleurs évident que sa mise en œuvre ajoute une complexité aux systèmes informatiques, complexité à la fois logicielle et matérielle. À partir de ce constat, on est obligé de poser la question de la sécurité informatique dans ce contexte où l'architecture des processeurs devient de plus en plus complexe, avec des modes de plus en plus privilégiés. Étant donné la complexité des systèmes

informatiques, l'exploitation de vulnérabilités présentes dans les couches privilégiées ne risque-t-elle pas d'être très sérieuse pour le système global ? Étant donné la présence de plusieurs machines virtuelles, qui ne se font pas mutuellement confiance, au sein d'une même machine physique, est-il possible que l'exploitation d'une de ces vulnérabilités soit réalisée par une machine virtuelle compromise ? N'est-il pas nécessaire d'envisager de nouvelles architectures de sécurité prenant en compte ces risques ?

Ainsi la sécurité informatique est un ensemble de moyens techniques, organisationnels, juridiques et humains nécessaires pour conserver, rétablir et garantir la sécurité du système d'information. Cela signifie que la sécurité doit être abordée dans un contexte global et notamment prendre en compte les aspects suivants :

- La sensibilisation des utilisateurs aux problèmes de sécurité
- La sécurité logique, c'est-à-dire la sécurité au niveau des données, notamment les données de l'entreprise, les applications ou encore les systèmes d'exploitation.
- La sécurité des télécommunications : technologies réseau, serveurs de l'entreprise, réseaux d'accès, etc.
- La sécurité physique, soit la sécurité au niveau des infrastructures matérielles : salles sécurisées, lieux ouverts au public, espaces communs de l'entreprise, postes de travail des personnels, etc.

On peut déduire de ces constats que la démarche de sécurité informatique est une activité managériale des systèmes d'information et qu'il convient aussi d'établir un tableau de bord de pilotage associé à une politique de sécurité comprenant les organes vitaux constituant une entreprise.

Le système d'information permet aux différents acteurs interne ou externe de véhiculer, de partager des informations et de communiquer grâce à un ensemble de ressources matérielles, humaines et logicielles. Ce qui expose ce système à de nombreuses menaces de diverses origines. Cependant, la sécurité du système d'information est aujourd'hui une préoccupation très importante d'une entreprise. Pour faire face à ce fameux problème de sécurité, dans cette thèse nous proposons des approches de solution de contrôle d'accès, de filtrage des paquets, de chiffrement de communication et de surveillance l'infrastructure qui héberge le système d'information.

Cette thèse est organisée en six chapitres :

- Dans le premier chapitre nous décrivons le principe, les objectifs, les domaines, et différentes facettes de la sécurité du système informatique. Ainsi, décrit les types, les méthodes, les outils d'attaques des réseaux informatiques.
- Dans le deuxième chapitre on donne un aperçu sur les systèmes contribuant à la sécurité des systèmes d'information. Dans ce chapitre nous présentons les techniques de cryptographie pour sécuriser le système d'information, et l'utilisation des protocoles de sécurité qui permet de sécuriser la circulation de l'information confidentielle sur le support de communication.
- Le troisième chapitre consacré au système de détection d'intrusion (IDS), pour connaitre les attaques qui menacent notre système d'informatique pour définir les règles de sécurités adéquat. En fin, on va implémenter le système de détection d'intrusion libre « SNORT »
- Dans le quatrième chapitre nous présentons les concepts de la virtualisation et de machines virtuelles ainsi que de différentes techniques de mise en œuvre associées à ces concepts.
- Le cinquième chapitre présente les problèmes de sécurités et les différentes menaces de l'infrastructure virtuelle. Dans ce chapitre, nous proposons une approche de solutions de

sécurité de cette infrastructure par la mise en place un IPS open source comme front gateway et le pile ELK comme une plateforme de surveillance.

- Dans le sixième chapitre, nous nous sommes concentrés sur les IDS basés sur les signatures, l'accent étant mis sur l'évaluation des performances dans les réseaux à grande vitesse. Notre objectif est fourni une comparaison détaillée entre les deux IDS (Snort et Suricata) avec un trafic à grande vitesse.

Nous terminons ce manuscrit par une conclusion générale et nous présentons quelques perspectives.

Chapitre 1 : Principes de la sécurité et attaques informatique

1.1 Introduction

La Sécurité des Systèmes d'Information (SSI) est aujourd'hui un sujet important parce que le système d'information (SI) est pour beaucoup d'entreprises un élément absolument vital. Puisque le SI est vital, tout ce qui le menace est potentiellement mortel. Conjurer les menaces contre le SI est devenu impératif, et les lignes qui suivent sont une brève description de ce qu'il faut faire pour cela.

Les menaces contre le système d'information entrent dans une des catégories suivantes : atteinte à la disponibilité des systèmes et des données, destruction de données, corruption ou falsification de données, vol ou espionnage de données, usage illicite d'un système ou d'un réseau, usage d'un système compromis pour attaquer d'autres cibles, etc.

Les menaces engendrent des risques et coûts humains et financiers : perte de confidentialité de données sensibles, indisponibilité des infrastructures et des données, dommages pour le patrimoine intellectuel et la notoriété. Les risques peuvent se réaliser si les systèmes menacés présentent des vulnérabilités. Il est possible de préciser la notion de risque en la décrivant comme le produit d'un préjudice par une probabilité d'occurrence [1] :

risque = préjudice x probabilité d'occurrence

Cette formule exprime qu'un événement dont la probabilité est assez élevée, par exemple la défaillance d'un disque dur, mais dont il est possible de prévenir le préjudice qu'il peut causer, par des sauvegardes régulières, représente un risque acceptable ; il en va de même pour un événement à la gravité imparable, comme l'impact d'un météorite de grande taille, mais à la probabilité d'occurrence faible.

Il va de soi que, dans le premier cas, le risque ne devient acceptable que si les mesures de prévention contre le préjudice sont effectives et efficaces : cela irait sans dire, si l'oubli de cette condition n'était très fréquent.

Si la question de la sécurité des systèmes d'information a été radicalement bouleversée par l'évolution rapide de l'Internet, elle ne saurait s'y réduire ; il s'agit d'un vaste problème ont les aspects techniques ne sont qu'une partie. Les aspects juridiques, sociaux, ergonomiques, psychologiques et organisationnels sont aussi importants, sans oublier les aspects immobiliers, mais nous commencerons par les aspects techniques liés à l'informatique.

1.2 Objectifs de sécurité des systèmes d'informations

La sécurité des systèmes d'information est un ensemble des moyens techniques, organisationnels, juridiques et humains mis en place [2], qui repose principalement sur les quatre critères suivants (Figure1) : disponibilité, intégrité, confidentialité et audibilité [1]

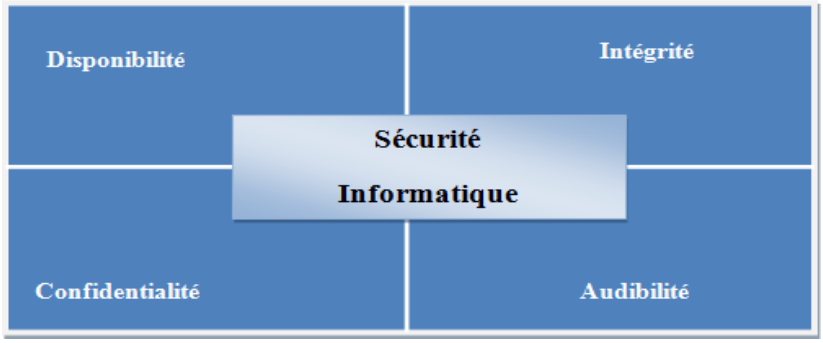

Figure 1 : Enjeux de la sécurité des systèmes d'information

- Disponibilité : c'est la garantie pour que les informations ou services soient accessibles et utilisable dans des conditions spécifiées de temps.
- Confidentialité : c'est la garantie que l'accès aux informations et services considérés ne soit pas possible sans autorisation.
- Intégrité : c'est la garantie que les informations ou services considères ne soient pas modifiés de manière indue.
- Audibilité : garantir la traçabilité des accès, des tentatives d'accès et la conservation de ces traces comme preuves exploitables

En général, la sécurité informatique consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées pour les fins auxquelles elles ont été conçues au début [2]. De plus elle vise à inscrire l'évolution des systèmes informatiques dans le cadre d'un processus d'amélioration continue.

1.3 Domaines de la sécurité

Tous les domaines de l'informatique et des réseaux de télécommunication sont concernés par la sécurité d'un système d'information. En fonction de son domaine d'application, la sécurité informatique se décline en :

Sécurité physique : Concerne tous les aspects liés à la maîtrise des systèmes et de l'environnement dans lesquels ils se situent. La sécurité physique repose essentiellement sur :

- des normes de sécurité (climatisation, alimentation électrique, etc.) ;
- la protection de l'environnement (incendie, inondation, température, à l'humidité, etc.) ;
- des mesures de gestion et de contrôle des accès physiques aux locaux, équipements et infrastructures ;
- la redondance physique des infrastructures et sources énergétiques ;
- le plan de maintenance préventive (tests, etc.) et corrective (pièces de rechange, etc.) des équipements ce qui relève également de la sécurité de l'exploitation des environnements.

Sécurité de l'exploitation : La sécurité de l'exploitation doit permettre un bon fonctionnement opérationnel des systèmes informatiques. Cela comprend la mise en place d'outils et de procédures relatifs aux méthodologies d'exploitation, de maintenance, de test, de diagnostic, de gestion des performances, de gestion des changements et des mises à jour. La sécurité de l'exploitation dépend fortement de son degré d'industrialisation qui est qualifié par le niveau de supervision des applications et l'automatisation des tâches. Quelques points clés de cette sécurité :

- Plan de sauvegarde, de secours, de continuité, de tests ;
- Inventaire réguliers et si possible dynamique ;
- Gestion du parc informatique, des configurations et des mises à jour ;
- Contrôle et suive de l'exploitation.

Sécurité logique : La sécurité logique fait référence à la réalisation de mécanismes de sécurité par logiciel contribuant au bon fonctionnement des programmes, des services offerts et à la protection des données. Elle repose sur la mise en œuvre d'un système de contrôle d'accès logique s'appuyant sur un service d'authentification, d'identification et d'autorisation. Elle repose également sur :

- Les dispositifs mis en place pour garantir la confidentialité la cryptographie ;
- Une gestion efficace des mots de passe et des procédures d'authentification ;
- Des mesures antivirus et de sauvegarde des informations sensibles.

Pour déterminer le niveau de protection nécessaire aux informations manipulées, une classification des données est à réaliser pour qualifier leur degré de sensibilité (normale, confidentielle, top secrète, ...)

La sécurité applicative : Faire un développement pertinent et l'intégrer harmonieusement dans les applications existantes. Cette sécurité repose essentiellement sur :

- Une méthodologie de développement ;
- La robustesse des applications ;
- Des contrôles programmés ;
- Des jeux de tests ;
- Un plan de migration des applications critiques ;
- La validation et l'audit des programmes ;
- Un plan d'assurance sécurité .

Sécurité des infrastructures informatique et de télécommunication : La sécurité des télécommunications consiste à offrir à l'utilisateur final et aux applications communicantes, une connectivité fiable de « bout en bout » [2]. Cela passe par la réalisation d'une infrastructure réseau sécurisée au niveau des accès au réseau et du transport de l'information et cela s'appuie sur des mesures architecturales adaptées, l'usage de plates-formes matérielles et logicielles sécurisées et une gestion de réseau de qualité. La sécurité des télécommunications ne peut à elle seule garantir la sécurité des informations. Elle ne constitue qu'un maillon de la chaîne sécuritaire car il est également impératif de sécuriser l'infrastructure informatique dans laquelle s'exécutent les programmes. Pris au sens large, cela comprend la sécurité physique et environnementale des systèmes (poste de travail de l'utilisateur, serveur ou système d'information (Figure 2)).

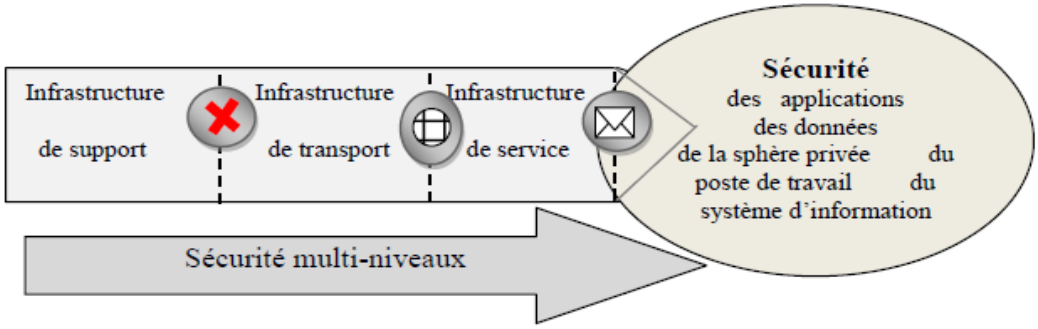

Figure 2 : Sécurité des infrastructures de télécommunication.

1.4 Différentes facettes de la sécurité

1.4.1 Diriger la sécurité

La sécurité informatique passe par la définition d'une politique de sécurité et la formation du personnel. Elle est en constante évolution et se traduit par un problème de gestion de la qualité constante lié à la maintenabilité et à l'évolution des systèmes, des enjeux et des risques

Dans de nombreuses entreprises, l'outil informatique est essentiel dans son développement, le moindre dysfonctionnement constitue donc un risque majeur.

1.4.2 Juridique

La responsabilité des acteurs (responsable sécurité, ...) est de plus en plus invoquée lors de sinistre où les ressources informatiques qu'ils gèrent sont l'objet ou le moyen d'une fraude.

Il est donc nécessaire de pouvoir prouver que des mesures sont pourtant prises pour sécuriser le système afin de se protéger contre un délit de manquement à la sécurité. Les responsables

d'entreprises doivent également être extrêmement attentifs à l'égard du droit des nouvelles technologies.

1.4.3 Architecture de sécurité

L'architecture de sécurité reflète l'ensemble des dimensions organisationnelle, juridique, humaine et technologique de la sécurité informatique à prendre en considération pour une appréhension complète de la sécurité d'une organisation (Figure 3). Définir une architecture globale de la sécurité permet de visualiser la dimension générale et la nature transversale de la sécurité informatique d'une entreprise et d'identifier ses diverses facettes et composantes afin de pouvoir les développer de façon cohérente, complémentaire et harmonieuse. Cela facilite l'intégration de mesures, de procédures et d'outils de sécurité.

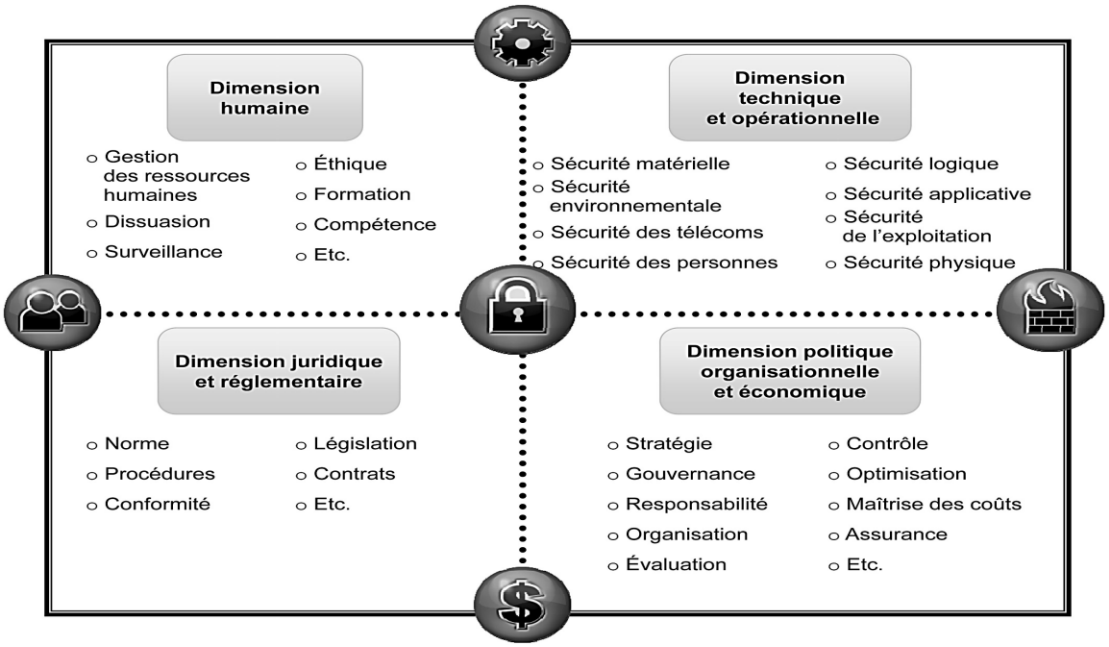

Figure 3 : Différentes dimensions d'une architecture de sécurité

Une démarche d'assurance des actifs, de gestion des risques, comme le respect des procédures, la formation, le comportement éthique des utilisateurs ou la conformité réglementaire sont autant de points à identifier dans un cadre d'architecture de sécurité. Ainsi, les critères de la sécurité pourront être réalisés judicieusement par le biais de mesures et de procédures complémentaires [29].

1.5 Attaques dans les réseaux informatiques

Une « attaque » est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables.

Sur internet des attaques ont lieu en permanence, à raison de plusieurs attaques par minute sur chaque machine connectée. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées (par des virus, chevaux de Troie, vers, etc.), à l'insu de leur propriétaire. Plus rarement il s'agit de l'action de pirates informatiques. Afin de contrer ces attaques il est indispensable de connaître les principaux types d'attaques afin de mettre en œuvre des dispositions préventives.

Les motivations des attaques peuvent être de différentes sortes :

- obtenir un accès au système ;
- voler des informations, tels que des secrets industriels ou des propriétés intellectuelles ;

- glaner des informations personnelles sur un utilisateur ;
- récupérer des données bancaires ;
- s'informer sur l'organisation (entreprise de l'utilisateur, etc.) ;
- troubler le bon fonctionnement d'un service ;
- utiliser le système de l'utilisateur comme « rebond » pour une attaque ;
- utiliser les ressources du système de l'utilisateur, notamment lorsque le réseau sur lequel il est situé possède une bande passante élevée

1.5.1 Classification des attaques

Il peut être utile de distinguer deux catégories d'attaques : les attaques passives et les attaques actives.

1.5.1.1 Attaques passives

Écoutes indiscrètes ou surveillance de transmissions sont des attaques de nature passive. Le but de l'adversaire est d'obtenir une information qui a été transmise. Ces attaques passives sont la capture du contenu d'un message et l'analyse de trafic. La capture du contenu de messages est facilement compréhensible. Une conversation téléphonique, un courrier électronique ou un fichier transféré peuvent contenir une information sensible ou confidentielle.

La seconde attaque passive, l'analyse de trafic, est plus subtile. Supposons qu'un moyen de masquer le contenu des messages ou des informations soit à disposition (par exemple, un système de chiffrement), de sorte que les adversaires, même en cas de capture, ne pourront en extraire l'information contenue. Cependant l'adversaire pourra être en mesure d'observer le motif de ces messages, déterminer l'origine et l'identité des systèmes en cours de communication, et observer la fréquence et la longueur des messages échangés. Cette information peut être utile pour deviner la nature de la communication. Les attaques passives sont très difficiles à détecter car elles ne causent aucune altération des données.

1.5.1.2 Attaques actives

Ces attaques impliquent certaines modifications du flot de données ou la création d'un flot frauduleux ; elles peuvent être subdivisées en quatre catégories : mascarade, rejoué, modification de messages et déni de service. Une mascarade a lieu lorsqu'une entité prétend être une autre entité. Une attaque de ce type inclut habituellement une des autres formes d'attaque active. Par exemple, des séquences d'authentification peuvent être capturées et rejouées, permettant ainsi à une entité autorisée munie de peu de privilèges d'en obtenir d'autres en usurpant une identité possédant ces privilèges. Le rejoué implique la capture passive de données et leur retransmission ultérieure en vue de produire un effet non autorisé.

La modification de messages signifie que certaines portions d'un message légitime sont altérées ou que les messages sont retardés ou réorganisés. Par exemple, le message " autoriser X à lire le fichier confidentiel comptes " est modifié en " autoriser Y à lire le fichier confidentiel comptes ". Le déni de service empêche l'utilisation normale ou la gestion de fonctionnalités de communication. Cette attaque peut avoir une cible spécifique ; par exemple, une entité peut supprimer tous les messages dirigés vers une destination particulière. Une autre forme de refus de service est la perturbation d'un réseau dans son intégralité, soit en mettant hors service le réseau, soit en le surchargeant de messages afin de dégrader ses performances.

1.5.2 Méthodes et types d'attaques

1.5.2.1 Attaque par intrusion(Sniffing)

Ce type d'attaque vise à s'infiltrer physiquement ou logiquement dans un système informatique en vue de récupérer des informations exploitables à d'autres fins.

Par exemple, installer un écouteur (sniffer) sur un réseau informatique constitue une attaque de type intrusion sur le réseau.

1.5.2.2 Attaque de l'homme du milieu (Man-In-The-Middle)

Ce type d'attaque vise à intercepter les communications entre deux parties (personne, ordinateur) sans que ni l'une, ni l'autre ne puisse s'en apercevoir. Il s'agit ici d'une attaque par interception. Le schéma ci-après illustre ce type d'attaque entre un ordinateur client et un ordinateur serveur (Figure 4)

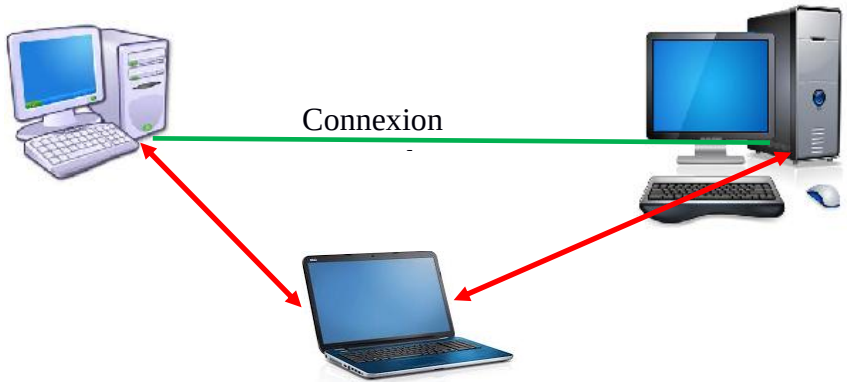

Figure 4 : Attaque de l'homme du milieu

1.5.2.3 Attaque par déni de service

Ce type d'attaque très fréquente, vise à perturber le bon fonctionnement d'un service (Figure 5). Elle exploite généralement les faiblesses de la pile de protocole TCP/IP et les vulnérabilités logicielles existantes et non traitées.

Exemple

Par exemple, envoyer 1.000.000 de requêtes en moins de 5 secondes à un serveur web avec des adresses IP sources fictives, constitue une attaque de type déni de service sur le serveur Web.

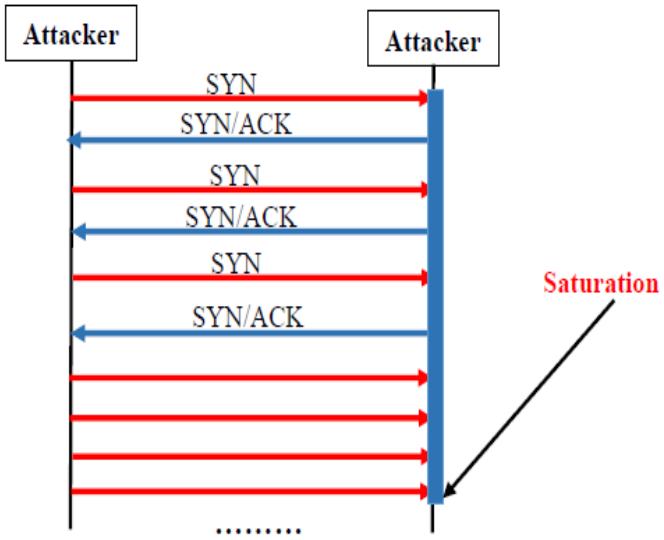

Figure 5 : Attaques Syn Flooding

1.5.2.4 Attaque par hameçonnage (phishing)

Le phishing est une technique dans laquelle des bandes organisées de cybercriminels se font passer pour des organismes financiers ou grandes sociétés en envoyant des emails ou des pages web frauduleux pour récupèrer des mots de passe de comptes bancaires ou numéros de cartes de crédit pour détourner des fonds. Le phénomène existe depuis 1996 et a connu une accélération significative début 2003.

1.5.2.5 Attaque par dictionnaire et par force brute (mot de passe)

- Attaque par dictionnaire

L'attaque par dictionnaire vise à retrouver un mot de passe à partir d'un dictionnaire élaboré au préalable. Elle consiste à tester une série de mots de passe potentiels, les uns à la suite des autres, en espérant que le mot de passe utilisé pour le chiffrement soit contenu dans le dictionnaire.

- Attaque par force brute

L'attaque par force brute a le même objectif que l'attaque par dictionnaire, sauf que la technique change. Elle consiste à tester une à une, toutes les combinaisons possibles. Cette attaque est difficile d'aboutir lorsque le mot de passe contient plus de caractères variés (majuscules, minuscules, chiffres, caractère spéciaux).

1.5.2.6 Attaque par ingénierie sociale

Ce type d'attaque très fréquente également, vise à récupérer des informations sensibles des utilisateurs en s'appuyant sur leur naïveté. Elle exploite l'abus de confiance faite par les utilisateurs du système d'information.

Par exemple un hacker qui se fait passer pour un technicien du support en appelant une secrétaire pour lui demander le mot de passe d'ouverture de session sur son poste. Il s'agit là d'une usurpation d'identité.

1.5.3 Outils d'attaque

Pour mener à bien les attaques sur les systèmes informatiques, les pirates utilisent des outils informatiques bien connus du domaine. Ces outils sont également utilisés par les administrateurs et spécialistes de la sécurité pour tester la robustesse de leurs systèmes d'information, généralement dans le cadre d'un audit de sécurité.

Parmi ces outils, nous pouvons citer :

- Les programmes malveillants (virus, ver, cheval de troie, logiciel espion [spyware]);
- Les scanners et sniffers;
- Les backdors (portes dérobées);
- Les spams (courriers indésirables).

1.5.3.1 Programmes malveillants

- Virus

Un virus informatique est un programme malveillant conçu pour se propager à d'autres ordinateurs en s'insérant dans des programmes légitimes appelés « hôtes». Il peut perturber plus ou moins gravement le fonctionnement de l'ordinateur infecté. Il peut se répandre à travers tout moyen d'échange de données numériques comme les réseaux informatiques et les CD/DVD, les clefs USB, etc.

- Ver

Un ver informatique est un programme malveillant qui se reproduit sur plusieurs ordinateurs en utilisant un réseau informatique comme Internet. Contrairement à un virus informatique, un ver n'a pas besoin d'un programme pour se reproduire. Il exploite les différentes ressources de l'ordinateur qui l'héberge pour assurer sa reproduction.

Exemple :
Comme exemple de ver, nous avons le ver "**I LOVE YOU**" découvert pour la première fois le 4 mai 2000, le ver "**MORRIS**" découvert en 1988. Ce ver a été à l'origine de la création du CERT (Computer Emergency Response Team).

- Cheval de troie

Un cheval de Troie est un programme d'apparence légitime conçu pour exécuter de façon cachée des actions à l'insu de l'utilisateur. En général, un cheval de Troie tente d'utiliser les droits appartenant à son environnement pour détourner, diffuser ou détruire des informations, ou encore pour ouvrir une porte dérobée qui permet à un attaquant de prendre, à distance, le contrôle de l'ordinateur.

- Logiciel espion (spyware, mouchard ou espiogiciel)

Un logiciel espion est un programme malveillant qui s'installe dans un ordinateur dans le but de collecter et transférer des informations sur l'environnement dans lequel il s'est installé, très souvent sans que l'utilisateur en ait connaissance. Un logiciel espion est généralement composé de trois mécanismes distincts : Infection, collecte et transmission

- Rootkit

Un rootkit ("jeu de démarrage" en français) est un programme malveillant dont la principale fonctionnalité est de dissimuler la présence de son activité et celle des autres programmes néfastes aux yeux de l'utilisateur du système et des logiciels de sécurité (antivirus, pare-feu, IDS).
Certains rootkit peuvent en plus de cette fonctionnalité principale, installer des backdors (porte dérobée).
Les rootkits ont deux caractéristiques principales :

- Ils modifient profondément le fonctionnement du système d'exploitation
- Ils se rendent invisibles (difficile à les détecter)

1.5.3.2 Sniffers

Un sniffer est un outil matériel ou logiciel, permettant de lire les données qui circulent dans un réseau. Si les données sont non chiffrées, on peut obtenir des informations sensibles comme les mots de passe (Figure 6). Ce genre d'outil peut également aider à résoudre des problèmes réseaux en visualisant ce qui passe à travers l'interface réseau.

Exemple
Un exemple d'outil sniffer : Wireshark

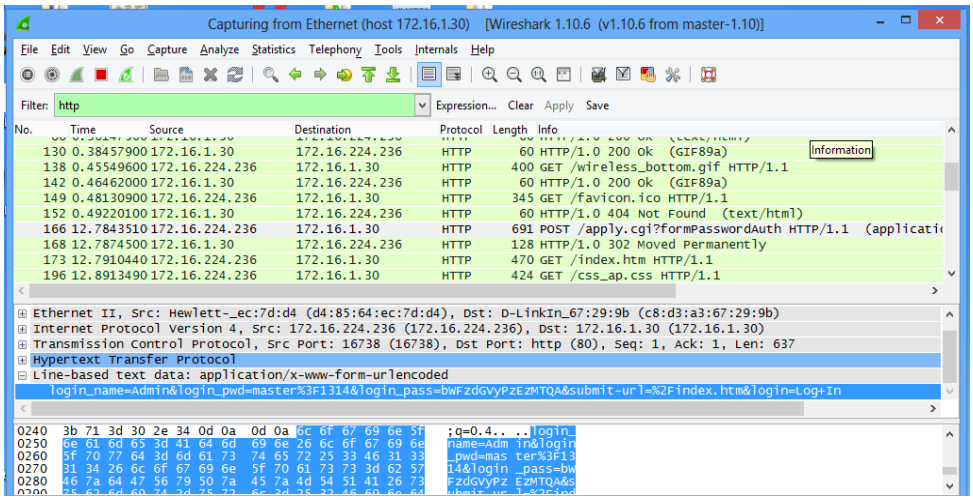

Figure 6 : Capture d'ecran Wireshark

1.5.3.3 ARP spoofing

L'ARP spoofing, ou ARP poisoning, est une technique utilisée en informatique pour attaquer tout réseau local utilisant le protocole de résolution d'adresse ARP, les cas les plus répandus étant les réseaux Ethernet et Wi-Fi. Cette technique permet à l'attaquant de détourner des flux de communications transitant entre une machine cible et une passerelle (Figure 7) : routeur, box, etc. L'attaquant peut ensuite écouter, modifier ou encore bloquer les paquets réseaux.

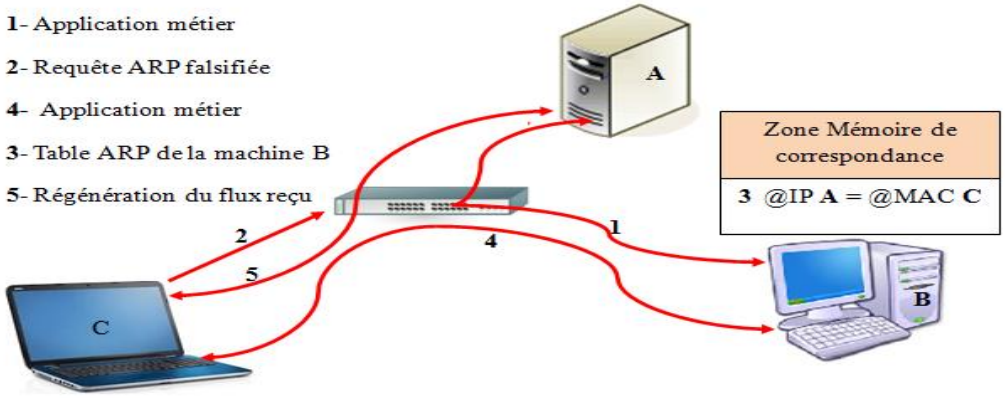

Figure 7 : ARP spoofing

Exemple :

Installation de l'outil dsniff (arpspoof)

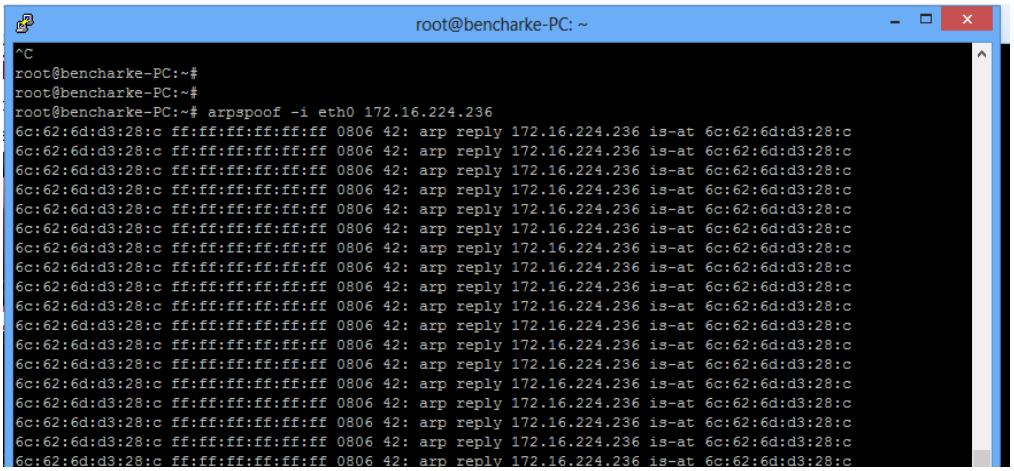

Figure 8 : Capture d'écran arp spoofing

1.5.3.4 Backdors (portes dérobées)

Une porte dérobée n'est pas un programme, mais une fonctionnalité d'un programme permettant de donner un accès secret au système. Ce genre de fonctionnalité est souvent ajouté à un logiciel par l'éditeur, afin de lui permettre de surveiller l'activité du logiciel, ou de prendre le contrôle en cas de sollicitation.

Généralement, les pirates informatiques une fois entrés dans le système, créent une porte dérobée afin de pourvoir y avoir accès à n'importe quel moment.

1.5.3.5 Spams (courriers indésirables)

Les spams sont des mails non sollicités qui envahissent nos boites mails. Généralement à caractère commercial, ils contiennent fréquemment des virus. Il est donc doublement utile de s'en protéger afin de ne pas être envahi par des données inutiles sur sa boite mail et afin d'éviter l'infection de son PC par un virus ou un spyware.

1.6 Conclusion

La sécurité des systèmes d'information représente aujourd'hui une tâche de fond à prendre en compte par toute entreprise qui désire disposer d'un ensemble d'outils et de méthodes qui lui permettent et assurent la gouvernance de son système d'information.

Bien évidement la sécurité à 100% reste un idéal à atteindre, surtout devant le large éventail des menaces qui mettent en danger l'exploitation d'un système d'information. Ainsi il est important de bien formaliser une politique de sécurité en prenant en compte les risques réelle qu'encourt un système informatique et en évaluant les coûts que peuvent engendrer les problèmes résultants de ces risques par rapport au coût nécessaire à la mise en place des solutions palliative à ces problèmes.

Chapitre 2 : Systèmes contribuant à la sécurité du système d'information

2.1 Introduction

Avec l'accroissement de la circulation de données et la confidentialité dans les réseaux, la sécurité des informations est devenue un problème crucial. Puisque, la fréquence des actions non autorisées dans les systèmes d'information (SI) augmente sans cesse, ce qui entraîne inévitablement d'énormes pertes financières et matérielles. Les incidents causés par des attaques sur les systèmes informatiques, liés à l'utilisation croissante des réseaux informatique, augmentent d'une manière exponentielle. Dans ce chapitre on va étudier les mesures de sécurité informatique (chiffrement, hachage et certificat) et les protocoles de sécurités.

2.2 Chiffrement

Le chiffrement consiste à rendre illisible un message en brouillant ses éléments de telle sorte qu'il soit très difficile de reconstituer l'original si l'on ne connaît pas la transformation appliquée.
Le chiffrement est basé sur deux éléments : **une clé** (une chaine de nombre binaire) et un **algorithme**. Les algorithmes peuvent être classés par leur caractère en deux types : chiffrement symétrique et chiffrement asymétrique.

2.2.1 Chiffrement symétrique

Le chiffrement symétrique utilise une clé unique partagée entre les 2 interlocuteurs. On encode et on décode le message avec la même clé (Figure 9), cette clé est appelé clé secrète [1].
Le problème de ce chiffrement est qu'il faut trouver un moyen de transmettre la clé secrète entre les 2 interlocuteurs.

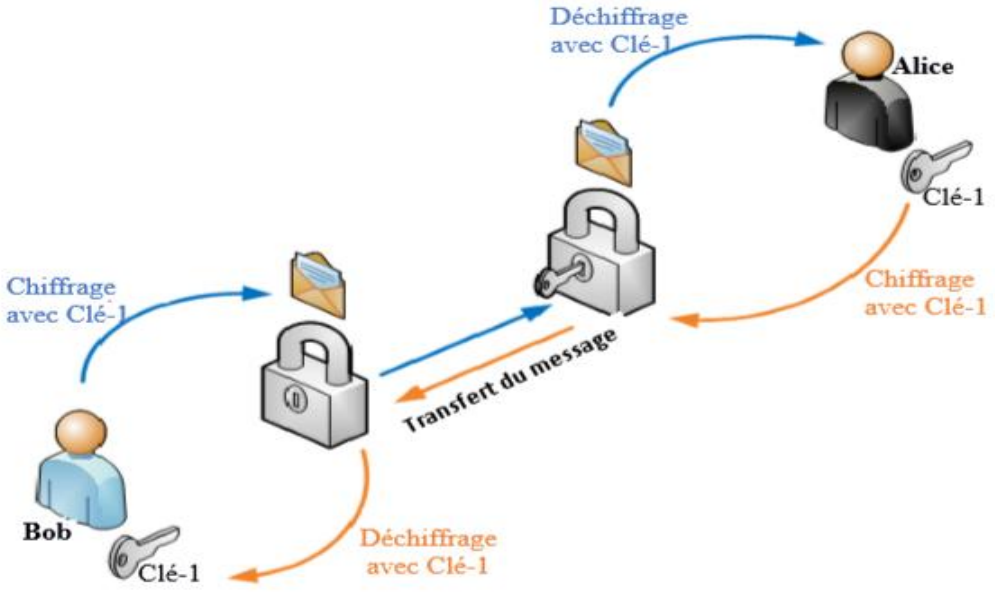

Figure 9 : Chiffrement symétrique

Il y a deux catégories de systèmes à clé privée : les chiffrements par blocs et les chiffrements de flux [3].

2.2.1.1 Chiffrement par flot « algorithme RC4 »

RC4 est un algorithme de chiffrement à flot [28] destiné aux applications logicielles. Il a été conçu par R. Rivest en 1987 pour les laboratoires RSA Malgré un certain nombre de faiblesses, il

est encore très utilisé aujourd'hui, notamment du fait de sa vitesse élevée. Il est employé par exemple dans SSL/TLS, protocole permettant d'assurer la confidentialité des transactions Web, dans la norme de chiffrement WEP (Wired Equivalent Privacy) pour les réseaux sans fil, IEEE 802.11b [4].

Principe de l'algorithme

RC4 se décompose en deux algorithmes, Le Key Scheduling Algorithm (KSA) et le Pseudo Random Generator Algorithm (PRGA) [4].

Algorithme KSA

L'objectif de cet algorithme est de mélanger un tableau appelé S contenant 256 octets à l'aide d'une clé secrète.

```
Algorithme KSA
Entre : la clef K secret de longueur L
pour i de 0 à 255
        S[i] := i
finpour
        j := 0
pour i de 0 à 255
        j := (j + S[i] + K[i mod L]) mod 256
échanger(S[i], S[j])
finpour
```

Algorithme PRGA

Une fois le KSA exécuté, nous obtenons un tableau de 256 valeurs permutées en fonction de la clé secrète. Ce tableau, va ensuite être utilisé par le PRGA pour générer un octet de keystream. Celui-ci sera ensuite XORé avec le texte clair pour obtenir le texte chiffré.

```
i := 0
j := 0
tant_que générer une sortie :
        i := (i + 1) mod 256
        j := (j + S[i]) mod 256
        échanger(S[i], S[j])
        octet_chiffrement = S[(S[i] + S[j]) mod 256]
        result_chiffré = octet_chiffrement XOR octet_message
fintant_que
```

Cet algorithme garantit que chaque valeur de S est échangée au moins une fois toutes les 256 itérations (Figure 10).

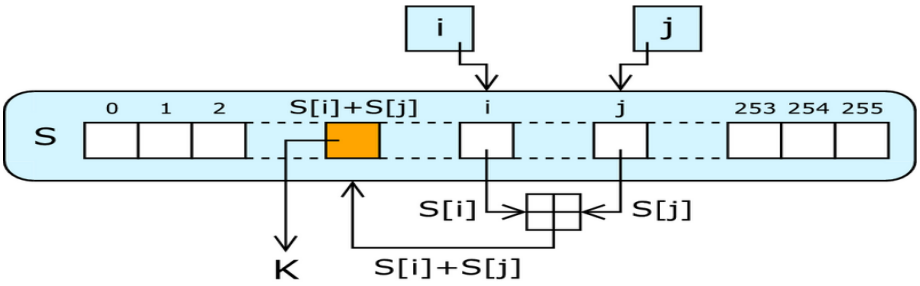

Figure 10 : Schéma de la génération d'un octet par RC4

2.2.1.2 Chiffrement par bloc «algorithme Blowfish»

Blowfish a été conçu par Bruce Schneier en 1993 [5] comme étant une alternative aux algorithmes existants, en étant rapide et gratuit. Blowfish est sensiblement plus rapide que le DES [6][7]. Il est un chiffrement Feistel, utilisant itérativement une fonction de chiffrement 16 fois. La grandeur des blocs est de 64 bits [5][7]. Il peut prendre une longueur de clé variant entre 32 bits et 448 bits [6][7]. Depuis sa conception il a été grandement analysé et est aujourd'hui considéré comme étant un algorithme de chiffrement robuste. Il n'est pas breveté et ainsi son utilisation est libre et gratuite. Il est utilisé dans de nombreux logiciels propriétaires et libres (dont GnuPG et OpenSSH).

Il y a deux parties dans l'algorithme : une première partie qui manipule l'expansion de la clé et une deuxième partie qui manipule le chiffrement des données.

Principe général

Le schéma montre la structure principale de Blowfish (Figure 11). Chaque ligne représente 32 bits. L'algorithme gère deux ensembles de clés : les 18 entrées du tableau P et les quatre S-Boxes de 256 éléments chacune.

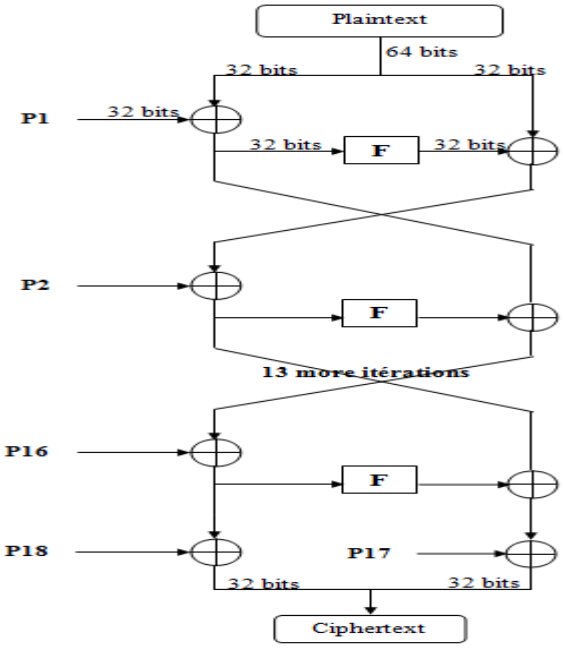

Figure 11 : Schéma de Feistel dans Blowfish

Les S-Boxes acceptent un mot de 8 bits en entrée et produisent une sortie de 32 bits. Une entrée du tableau P est utilisée à chaque tour. Arrivé au tour final, la moitié du bloc de données subit un XOR avec un des deux éléments restants dans le tableau P [5].

La fonction de chiffrement F

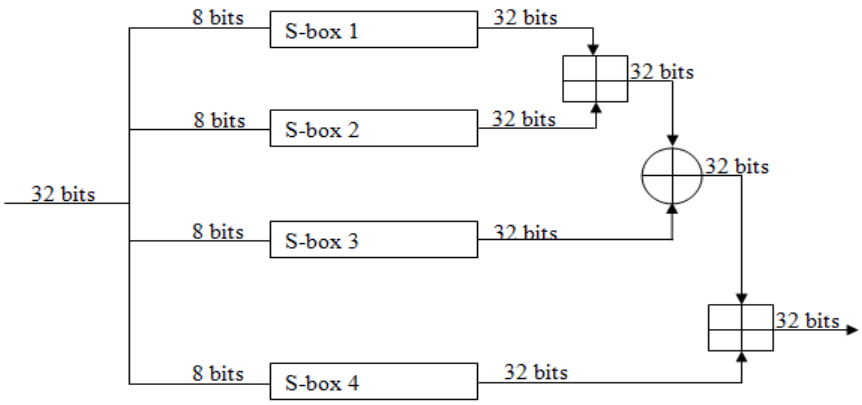

Figure 12 : F-fonction de Blowfish

La fonction F de Blowfish sépare une entrée de 32 bits en quatre morceaux de 8 bits et les utilise comme entrées pour accéder aux S-Boxes[7]. Les sorties sont additionnées avec une somme modulo 232 et l'algorithme effectue un XOR entre les deux sous-totaux pour produire la sortie finale de 32 bits.

En tant que schéma de Feistel, Blowfish peut être inversé simplement en appliquant un XOR des éléments 17 et 18 du tableau P sur le bloc chiffré. Il faut ensuite utiliser les entrées du tableau P dans l'ordre inverse.

Initialisation de la structure

La préparation de la structure à partir de la clé commence avec l'initialisation du tableau P et des S-Boxes avec des valeurs qui sont tirées du nombre Pi exprimé en hexadécimal [6].

On opère ensuite un XOR entre la clé secrète et les entrées du tableau P (avec une extension cyclique de la clé si nécessaire). Un bloc de 64 bits, tous à zéro, est ensuite chiffré avec cette version temporaire de Blowfish. Le résultat chiffré remplace ensuite le premier et le deuxième élément du tableau P.

On réitère l'opération de chiffrement avec cette nouvelle version et ceci sur le résultat précédent. On obtient alors le troisième et quatrième élément de P. L'algorithme continue ainsi en remplaçant tout le tableau P et les éléments des S-Boxes.

Au final, environ 4 Ko de données doivent être générées et Blowfish effectue 512 itérations pour y parvenir. De part ces contraintes, Blowfish est lent quand il faut changer de clé mais très rapide pour le chiffrement pris séparément [6]**.**

Algorithme Blowfish

Note : l'entrée de 64 bits de texte clair est notée "x" et le tableau P de 18 clés est noté P[i], où "i" est l'itération.

```
Début chiffrement
        Divisé x en 2 : xL et xR
        Pour i allant de 1 à 16 faire
                xL = xL XOR P[i]
                xR = F(xL) XOR xR
                Permuter xL et xR
        Fin Pour
```

Permuter xL et xR
xR = xR XOR P[17]
xL = xL XOR P[18]
x = xL + xR
Retourner x
Fin chiffrement

La fonction F de l'algorithme :

La fonction F( )
Début fonction F(Entrée : xL : 32 bits de données)
Divisé xL en 4 : a, b, c, d
Retourner ((S1,a + S2,b MOD 2^{32}) XOR S3,c) + S4,d MOD 2^{32}
Fin fonction F

2.2.2 Chiffrement asymétrique

Après avoir vu le chiffrement symétrique, nous allons voir le fonctionnement de chiffrement asymétrique, qui fonctionne par paire, contenant une clé publique et une clé privée (Figure 13). La clé publique étant distribuée aux personnes qui doivent chiffrer des messages pour la personne détenant la clé privée. Une clé privée, qui, comme son nom l'indique doit rester privée puisque les messages sont déchiffrables uniquement grâce à elle [8].

Pour résumer, la clé publique sert à chiffrer les messages et la clé privée à la déchiffrer. Ceci ne marche donc que dans un sens, on chiffre des messages pour quelqu'un d'unique qui pourra les ouvrir. Pour que la «conversation» puisse s'effectuer de manière bidirectionnelle, il faut générer deux paires de clés : chacun garde sa clé privée et distribue sa clé publique à la partie émettrice.

Principe de chiffrement asymétrique

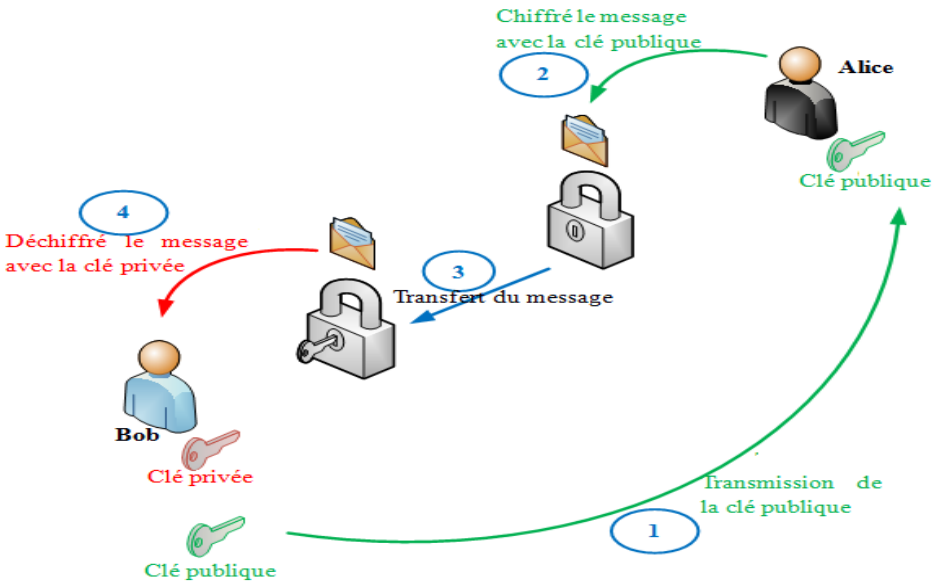

Figure 13 : Principe de chiffrement asymétrique

A travers ce schéma on voit que Bob possède deux clés, une clé privée et une clé publique, ce sera lui qui recevra les messages et qui sera le seul à pouvoir les déchiffrer. Dans un premier temps, Bob transmet sa clé publique à Alice pour qu'il puisse lui transmette des messages chiffrés. Ensuite, Alice chiffre un message avec la clé publique reçu puis transfert le message à Bob. Pour finir, Bob déchiffre le message grâce à sa clé privée qui est la seule à pouvoir déchiffrer le message.
Les systèmes à clé publique les plus utilisés sont RSA et ALGAMAL.

Algorithme RSA

Inventé en 1978 par Ronald L. Rivest, Adi Shamir et Leonard M. Adleman, le RSA est le plus populaire des systèmes à clé publique. Il peut être utilisé pour chiffrer des informations et/ou pour les signer (signature numérique) [9].

Initialisation

1- choisir deux grands nombres premiers, p et q.
2- Calculer n = p · q
3- Choisir e aléatoire tel que e et ((p - 1) · (q - 1)) n'aient aucun facteur commun excepté 1.
4- Trouver d tel que (e · d) soit divisible par ((p - 1) · (q - 1)), donc : ed = 1 mod((p - 1)(q - 1)).

Clé publique : (n,e).
Clé privée : (n,d) ou (p,q,d) si on désire garder p et q

Chiffrement/Déchiffrement

1- L'expéditeur crée le texte chiffré **c** à partir du message **m** :
C = $\mathbf{m^e}$ **mod(n)**, où **(n,e)** est la clé publique du destinataire.
2- Le destinataire reçoit **c** et effectue le déchiffrement :
m = $\mathbf{c^d}$ **mod(n)**, où **(n,d)** est la clé privée du destinataire.

Signature numérique

1- L'expéditeur crée la signature **s** à partir du message **m** :
s = $\mathbf{m^d}$ **mod(n),** où **(n,d)** est la clé privée de l'expéditeur.
2- Le destinataire reçoit **s** et **m** et effectue la vérification de **m** :
m = $\mathbf{s^e}$ **mod(n),** où **(n,e)** est la clé publique de l'expéditeur

2.2.3 Chiffrement hybride

Cette technique a été introduite afin de profiter des avantages des deux techniques précédemment citées, c'est à dire la rapidité de traitement des messages codés par cryptographie symétrique et la puissance du chiffrement des messages par cryptographie asymétrique.
Le principe est assez simple. La communication entre A et B se fait par système cryptographique symétrique, ce qui rend la communication assez rapide à chiffrer et déchiffrer. Mais la lacune de la sécurité de transmission de la clé symétrique de chiffrement/déchiffrement est palliée par un chiffrement de cette clé, qui lui est asymétrique (Figure 14).

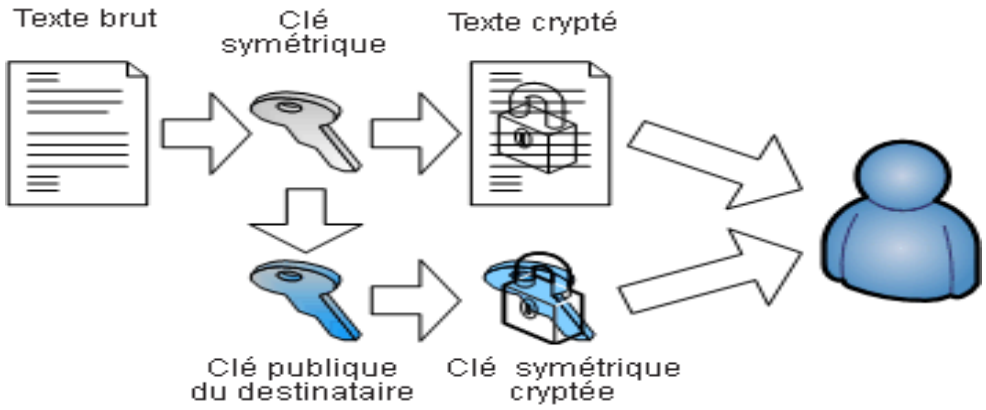

Figure 14 : Principe de cryptage hybride

2.2.4 Comparaison entre quelque algorithmes de chiffrement

Algorithme	Longueur de clé (bits)	Vitesse (Mo/s)
RC4	128	113.35
DES	56	22.19
3DES	168	9.8
AES	128	62.02
RSA	1024	0.02

Table 1 : Table de comparaison des vitesses des algorithmes de chiffrement [9]

2.3 Signature numérique et authentification

2.3.1 Principe

La signature numérique permet d'identifier et d'authentifier l'expéditeur des données. Elle permet en outre de vérifier que les données transmises sur le réseau n'ont pas subi de modification.
Le principe de signature numérique (Figure 15) est de hacher (condenser) le message émis par l'émetteur. Du côté du récepteur, ce message est vérifié par la même fonction de hachage. Une fonction spéciale de hachage H est appliquée sur le message M de longueur variable. Le message H(M), la longueur est fixe, est l'empreinte du message original M. Cet empreinte est ensuite chiffré par la clé privé de l'expéditeur et devient E(H(M)). L'ensemble (M, E(H(M)) est envoyé au destinataire [9]. A la réception de l'ensemble (M, E(H(M)) le destinataire extrait E(H(M)) puis le déchiffre avec la clé publique de l'expéditeur pour avoir H'(M). Il applique la même fonction H sur le message M pour avoir H(M) et compare H(M) et H'(M). Si les deux sont identiques, l'intégrité du message et l'authentification de l'expéditeur sont vérifiées [9].

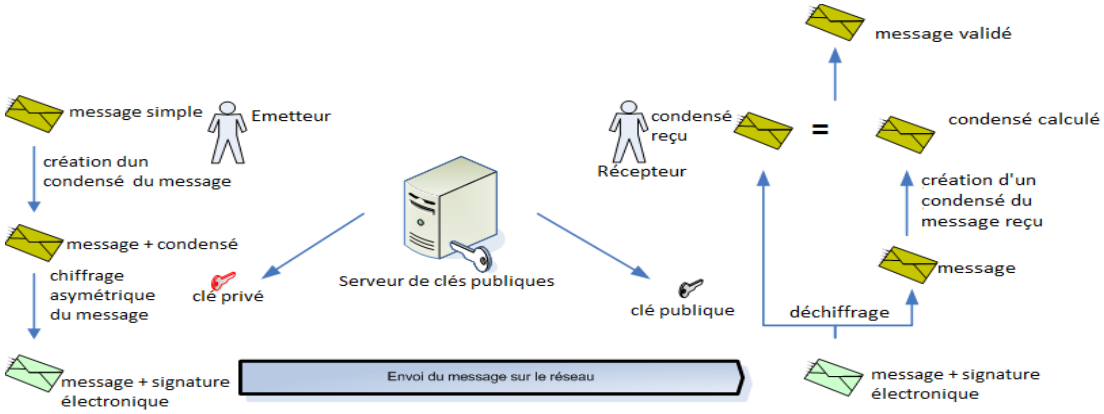

Figure 15 : Principe de signature numérique

2.3.2 Fonction de hachage

Une fonction de hachage construit une empreinte d'une chaine de données, à partir de laquelle il est impossible de revenir à la chaine de données initiale. La probabilité que deux chaines de données aient la même empreinte est très faible.

Cette fonction est utilisée pour la vérification de l'intégrité des messages transmis. On créer pour cela une empreinte du message à transmettre, puis on transmet le message et l'empreinte. A la réception du message on calcule l'empreinte du message reçu et on la compare à l'empreinte initiale. Si les deux empreintes correspondent, c'est que le message n'a pas pu être modifié [10].

Les principales fonctions de hachage sont :

- MD5 (Message Digest 5) : Mis au point par Ronal Rivest en 1991, MD5 découpe un message quelconque en blocs de 512 bits et calcule une empreinte de 128 bits [9].
- SHA-1 (Secure Hash Algorithm) est une fonction de hachage cryptographique conçue par la National Security Agency des États-Unis (NSA), et publiée par le gouvernement des États-Unis comme un standard fédéral de traitement de l'information . Elle produit un résultat (appelé « hash » ou condensat) de 160 bits.
- SHA-2 est, comme SHA-1, développé par la NSA, pour boucher les failles de la génération précédente de fonctions. Une particularité est que cette famille de fonctions de hachage va donner des empreintes de taille différentes, correspondant au suffixe de la fonction utilisée : SHA-256 va donner une somme de contrôle de 256 bits, SHA-512 fournira lui 512 bits, et ainsi de suite [11].

Le tableau suivant résume les caractéristiques de MD5 et de SHA-1 [9].

Caractéristiques	**MD5**	**SHA-1-160**
Longueur d'empreinte	128 bits	160 bits
Unité de traitement	512 bits	512 bits
Nombre de d'étape	64	80
Fonction logique	4	4
Vitesse (Mo/s)	204,55	72.60

Table 2 : Table de comparaison de MD5 avec SHA-1

2.3.3 Code d'Authentification du Message haché (HMAC)

Les fonctions de hachage, telles que SHA, ne sont pas conçues pour mettre en place une authentification forte car elles ne sont pas basées sur une clé secrète. L'algorithme HMAC incorpore une clé secrète dans la fonction de hachage (Figure 16). Il a été proposé comme RFC 2104 et a été choisi comme modèle pour l'implémentation de signatures numériques (MAC) garantissant la sécurité des messages IP ainsi que dans d'autres protocoles Internet comme TLS (Tansport Layer Security) [12].

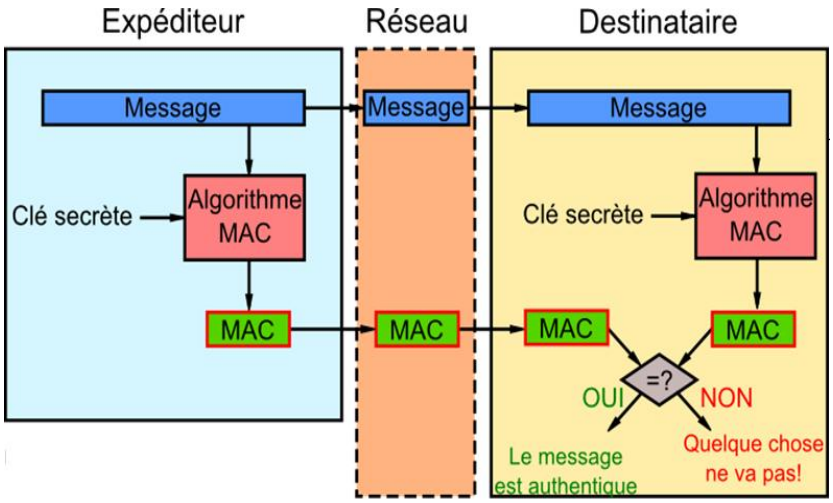

Figure 16 : Utilisation de l'algorithme HMAC

La fonction HMAC est définie comme suit :

$$\mathbf{HMAC_K(M)=H[(K\oplus opad)||H[(K\oplus ipad)||M]]}$$

Avec :

- H : une fonction de hachage itérative,
- K : la clé secrète complétée avec des zéros pour qu'elle atteigne la taille de bloc de la fonction H
- M : le message à authentifier,
- "||" : désigne une concaténation et "⊕" un « ou » exclusif,
- ipad et opad, chacune de la taille d'un bloc, sont définies par : ipad = 0x363636...3636 et opad = 0x5c5c5c...5c5c. Donc, si la taille de bloc de la fonction de hachage est 512, ipad et opad sont 64 répétitions des octets, respectivement, 0x36 et 0x5c [13].

2.4 Protocoles de sécurité

Dans le domaine de la sécurité dans les réseaux informatique, il existe plusieurs solutions de sécurité telles que le protocole IPSec (IP Security), le protocole SSL/TLS [9] (Socket Secure Layer/Transport Layer Security) et le protocole SSH (Secure Shell). Actuellement, il ne s'agit plus uniquement de chercher à faire de nouveaux développements afin que le réseau soit plus fiable ou d'une manière générale soit plus sécurisé dans son fonctionnement global. Il faudrait adapter ces solutions aux besoins spécifiques des utilisateurs ainsi qu'à leurs environnements.

2.4.1 Protocole SSH (Secure Shell)

2.4.1.1 Introduction

SSH, le Shell sécurisé, est développé en 1995 par Tatun Ylönen [25], un professeur finlandais. La première version de ce protocole, SSH-1, avait pour but de sécuriser les communications distantes au serveur Unix, surtout les commandes à distances (rcp, rlogin, telnet, rsh, ftp, …).

A cause de plusieurs failles de sécurité dans la première version de ce protocole, l'IETF a mis en place un groupe de travail appelé SECSH (Secure SHell) pour normaliser le protocole et orienter son développement dans l'intérêt du public. Ce groupe a soumis un ensemble de Drafts détaillant une nouvelle version décisive du protocole SSH 2.0. Cette version contient de nouveaux algorithmes et services comme le transfert des fichiers (SFTP) et le « tunneling » ou « port forwarding» des protocoles [25].

2.4.1.2 Architecture

SSH utilise une architecture client-serveur pour assurer l'authentification, le chiffrement et l'intégrité des données transmises dans un réseau. La version 2 de ce protocole spécifie une architecture composé en trois sous protocoles (ou couches) travaillant ensemble (Figure 17) :

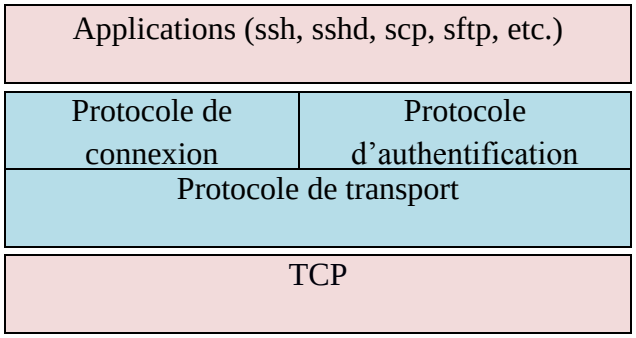

Figure 17 : Architecture SSH

- La couche transport SSH (SSH-TRANS) : Elle fournit l'authentification du serveur, la confidentialité et l'intégrité des données. Cette couche doit être créée pour que le client sache qu'il communique bien avec le bon serveur. Ensuite, la communication est chiffrée entre le client et le serveur au moyen d'un chiffrement symétrique.
- La couche d'authentification SSH (SSH-AUTH) : permet de certifier l'identité du client auprès du serveur. Cette couche est sécurisée par la clé de chiffrement créée dans la partie précédente. Après l'authentification du client auprès du serveur, de nombreux services différents peuvent être utilisés de façon sécurisée au cours de la connexion, tels qu'une session Shell interactive, des applications X11 et des ports TCP/IP [RFC761] tunnellisés.
- La couche de connexion SSH (SSH-CONN) : s'appuie sur la couche d'authentification. Elle offre une variété de services riches aux clients, en se servant de l'unique tube fourni par SSH-TRANS. Ces services comprennent tout ce qu'il faut pour gérer plusieurs sessions interactives ou non : multiplexage de plusieurs flux (ou canaux), gestion des transferts , de port et d'agent, etc. Même si SSH, dans sa version 2, est devenu plus modulaire, nous pouvons remarquer que les fonctionnalités de chaque sous protocole dépendent fortement des autres sous protocoles. Par exemple, le service d'authentification entre deux entités est divisé entre la couche d'authentification et la couche de transport [25].

Phase d'initialisation

Dans cette section, nous présentons sous forme d'étapes, la phase d'initialisation du protocole SSH (Figure 18). Dans la phase d'initialisation du protocole SSH, de nombreux éléments importants sont négociés. Dans le cas d'utilisation du protocole TCP/IP sur le réseau, la procédure suivante se déroule entre la machine cliente et la machine serveur :

1. Dès que la connexion est établie (protocole TCP sur le port 21 du serveur), le client et le serveur échangent en clair leurs numéros de version du protocole SSH. Sitôt ce premier échange est effectué, le client et le serveur utilisent un protocole binaire par paquet.
2. Le serveur commence par envoyer au client la liste des méthodes de cryptage supportées, la liste des méthodes d'authentification supportées, des indicateurs d'extensions de protocole (par exemple, la méthode de compression, etc.) et un cookie sur 64 bit que le client devra renvoyer. Ce cookie a pour but de protéger le serveur contre une attaque par déni de service.
3. Le client envoie à son tour la liste des méthodes de cryptage supportées, la liste des méthodes d'authentification supportées et une copie du cookie du serveur.

4. Le client et le serveur choisissent les meilleurs algorithmes supportés par les deux.
5. Le client et le serveur calculent séparément un identifiant de session à partir des valeurs Diffie-Hellman échangées entre les deux entités. SSHv2 utilise aussi une méthode d'échange des groupes DH (par exemple, diffie-hellman-group1-sha1) pour simplifier l'échange DH.
6. Le serveur envoie sa clé publique au client et signe avec sa clé privée les valeurs échangées précédemment.
7. Le client vérifie la signature du serveur et passe ensuite en mode crypté.
8. Le serveur répond au client par un message de confirmation crypté.
9. Les deux parties, le client et le serveur, sont maintenant en mode crypté avec utilisation de l'algorithme et de la clé sélectionnés.
10. Le client envoie maintenant la demande d'un service (par exemple, ssh-userauth ou ssh-connection).
11. Le serveur précise les méthodes d'authentification qu'il peut accepter (publickey, password, etc.).
12. Finalement, le client envoie sa méthode d'authentification que le serveur accepte ou rejette. Dans la plupart des implémentations SSH et suivant la politique du serveur, le client a le droit à trois essais pour s'authentifier.

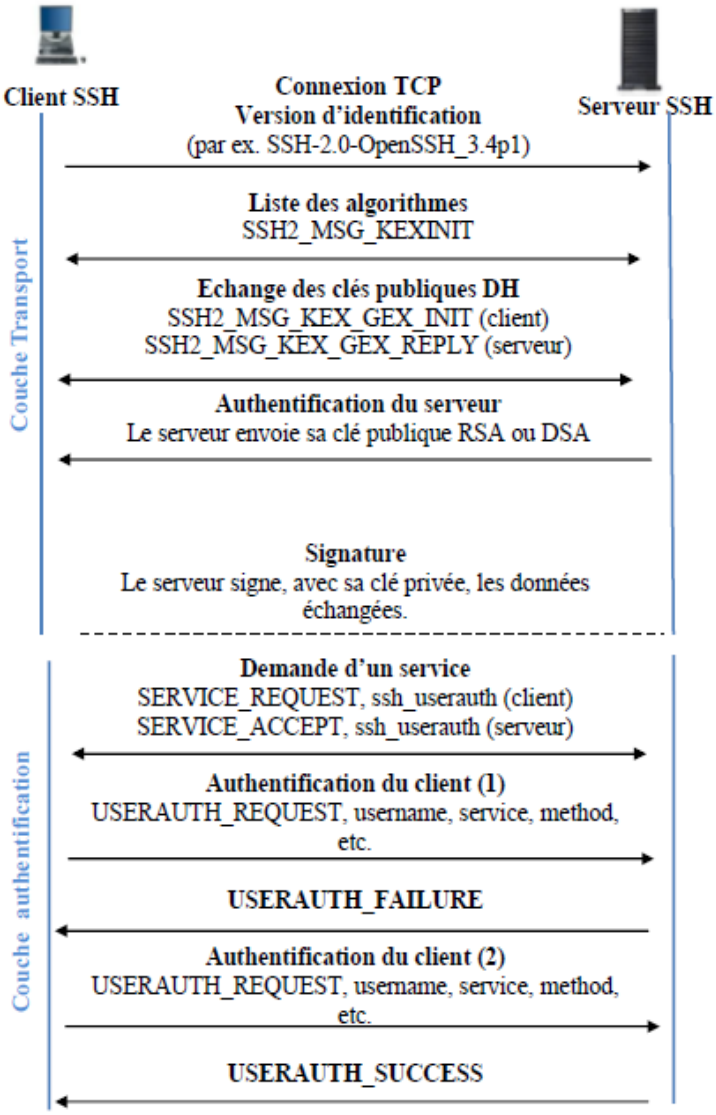

Figure 18 : Handshake de SSH

2.4.1.3 Avantages

SSH constitue une approche puissante et pratique pour protéger les communications sur un réseau d'ordinateurs. A travers son mécanisme d'authentification, SSH permet d'effectuer sous un tunnel sécurisé des connexions à distance, des transferts de fichiers et d'autres fonctionnalités importantes. Ainsi, il présente comme avantages :

- La création d'un VPN au niveau d'échange ;
- La notion du Transfert (tunneling) ;
- Le service de Single Sign On (SSO).

2.4.1.4 Inconvénients

SSH est capable de contourner de nombreuses menaces de sécurité liées au réseau.
Cependant, il est vulnérable aux attaques par déni de service, héritant ainsi les faiblesses de TCP/IP sur lequel il repose. En outre et suivant l'environnement, SSH est sensible à certaines méthodes d'attaques, comme l'analyse et le détournement de trafic. Les principaux inconvénients de SSH :

- La première authentification non sécurisée du client avec le serveur SSH ;
- Attaques sur le mot de passe ;
- Authentification non sûre par « hôte de confiance » ;
- SSH et la traduction des adresses réseau (NAT).

2.4.2 Protocole sécurisé SSL

2.4.2.1 Introduction

Le protocole SSL (Secure Socket Layer) a été développé par la société Netscape Communications Corporation [25] pour permettre aux applications client/serveur de communiquer de façon sécurisée. TLS (Transport Layer Security) est une évolution de SSL réalisée par l'IETF.
Conçu par Netscape, SSL est un protocole se plaçant entre la couche application et la couche transport permettant d'assurer la confidentialité, l'authentification et l'intégrité des données lors des communications.

2.4.2.2 Fonctionnement

La version 3 de SSL est utilisée par les navigateurs tels Netscape et Microsoft Internet Explorer depuis leur version 4. SSL est un protocole qui s'intercale entre TCP/IP et les applications qui s'appuient sur TCP. Une session SSL se déroule en deux phases :

1. Une phase de poignée de mains (handshake) : Durant cette phase le client et le serveur s'authentifient, par l'échange des certificats entre eux (Figure 19). Ensuite se mettent d'accord sur la fonction de hachage, l'algorithme de compression, l'algorithme de chiffrement et une clé pour chiffrer les messages qui seront appliqués dans l'étape suivante [9].

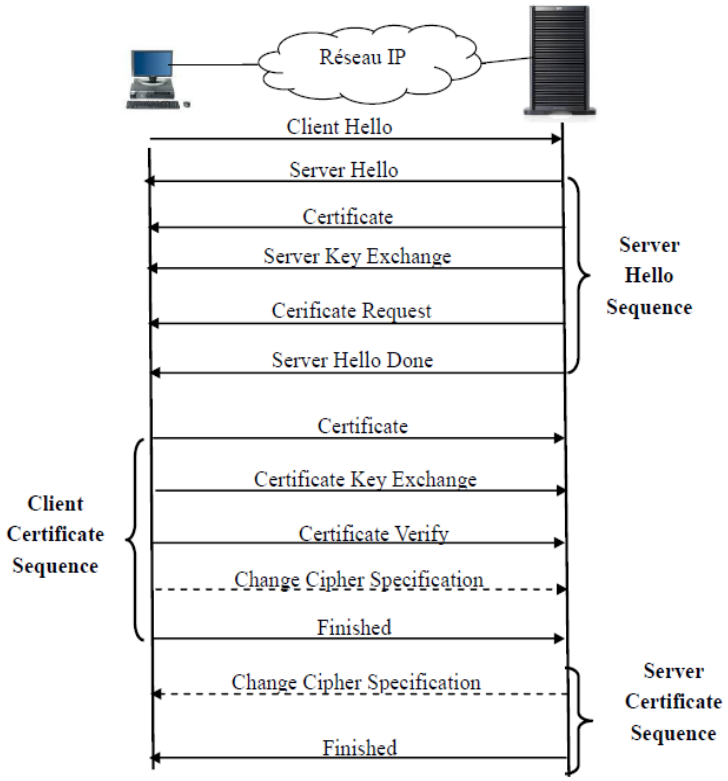

Figure 19 : Phase d'authentification du Protocole SSL

2. Une phase de communication durant laquelle les données sont transmis sous la forme d'une série de fragments d'une taille maximale de 16 Ko [9], chaque fragment étant protégé et transmis de façon individuelle. Pour transmettre un fragment, on commence par calculer son MAC (Message Authentication Code). La concaténation du fragment et de son MAC est cryptée, obtenant ainsi une charge cryptée (encrypted payload en anglais). On lui attache un en-tête, obtenant ainsi un enregistrement SSL (Figure 20)

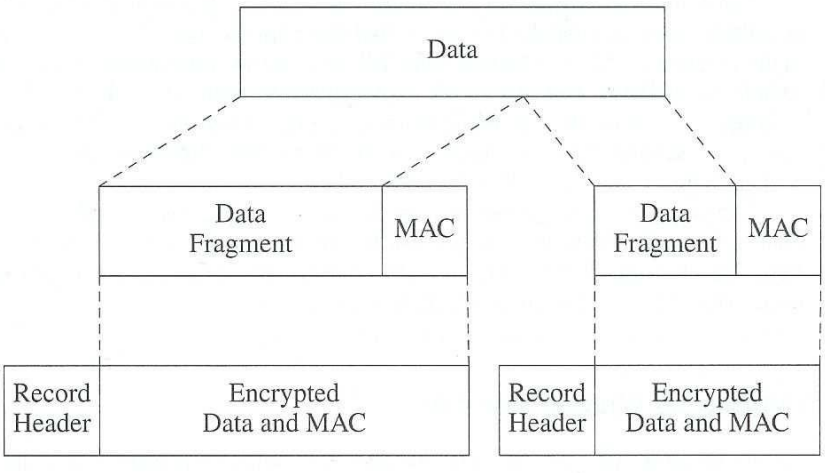

Figure 20 : Transmission de données avec SSL

2.4.2.3 Avantages

SSL/TLS est un protocole de sécurité qui permet de sécuriser les échanges entre un client et un serveur SSL/TLS d'une manière transparente, en se plaçant entre les couches d'application et de transport. Les principaux avantages de ce protocole sont :

- Authentification et intégrité fortes des messages ;
- Interopérabilité et facilité de déploiement ;
- Facilité d'utilisation ;
- SSL permet de créer des VPN.

2.4.2.4 Inconvénients

Le protocole SSL/TLS, comme toute autre technologie novatrice dans le domaine de sécurité, possède son lot d'inconvénients :

- Problèmes liés aux navigateurs;
- Problème cryptographique (lié à l'implémentation);
- Attaques liées au réseau ;
- HTTPS et les Proxy ;
- SSL et les hôtes virtuels ;
- Utilisation des faibles clés de chiffrement ;
- SSL/TLS et les applications basées sur UDP .

2.4.3 Protocole sécurisé au niveau réseau IPSEC

2.4.3.1 Introduction

IPSec (IP Security) est une suite des protocoles développé par l'IETF [25] (Internet Engineering Task Force) en 1992, destinés à sécuriser le trafic au niveau d'IP (IPv4 ou IPv6). Les services de sécurité offerts sont l'intégrité, l'authentification, la protection contre le rejeu et la confidentialité. Optionnel dans IPv4, IPsec est obligatoire pour toute implémentation d'IPv6 [14].

IPsec comporte deux modes, le mode transport qui protège juste les données transportées et le mode tunnel qui protège en plus l'entête.

2.4.3.2 Mécanismes de sécurité AH et ESP

Les services de sécurité offerts par IPsec reposent sur deux mécanismes, AH (Authentication Header) et ESP (Encapsulating Security Payload). Les paramètres nécessaires à l'utilisation de ces protocoles sont gérés à l'aide d'associations de sécurité (Security Association SA).

2.4.3.2.1 AH (Authentication Header)

Le protocole AH (Authentication) est conçu pour assurer l'intégrité et l'authentification des datagrammes IP sans chiffrement des données, c.à.d. sans confidentialité. Le principe d'AH est d'adjoindre au datagramme IP origine un champ supplémentaire permettant à la réception de vérifier l'authenticité des données incluses dans le datagramme.

L'authentification est basée sur l'utilisation d'un code d'authentification MAC (Message Authentification Code), soit SHA-1, soit MD5. Par conséquent, les deux parties communicantes doivent partager une clé secrète. L'entête d'authentification (Figure 21), comprend les champs suivants [9] :

- Suivant : identification du type d'entête suivant cet entête ;
- Longueur : longueur d'entête d'authentification ;
- Réserve : usage futur ;
- Index de paramètres de sécurité (SPI) : identification d'une association de sécurité ;

- Numéro de séquence : Numéro de séquence de 32 bits auto incrémenté, employé pour se protéger d'une retransmission du paquet;
- Authentification : champ variable de longueur multiple de 32 bits. Ce champ contient la valeur du MAC.

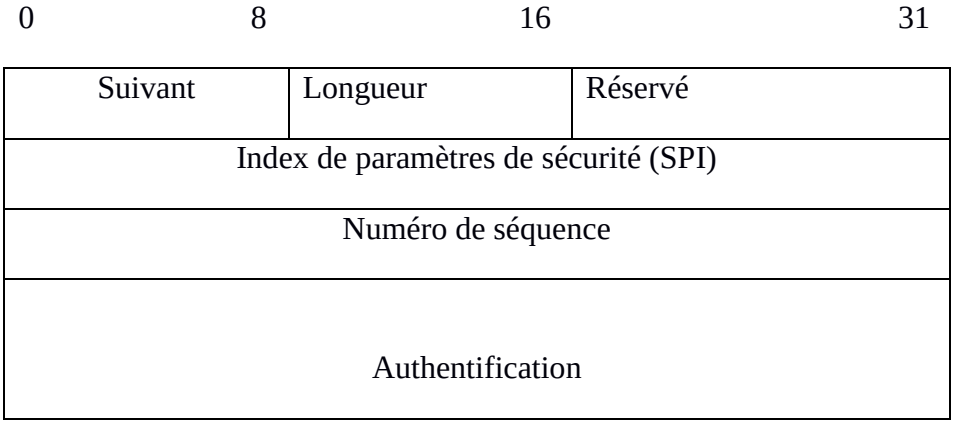

Figure 21 : Entête AH d'IPsec

2.4.3.2.2 ESP (Encapsulating Security Payload)

Le protocole ESP peut assurer, au choix, un ou plusieurs des services de sécurité suivants :

- la confidentialité des données par l'utilisation d'un système de chiffrement;
- l'authentification du paquet et de son émetteur (l'adresse source du paquet est celle de l'émetteur);
- l'intégrité des données (aucune altération volontaire ou non du paquet durant le transport) et l'unicité du paquet (pas de rejeu).

Contrairement au protocole AH, qui ajoute un entête supplémentaire au paquet IP, le protocole ESP fonctionne suivant le principe de l'encapsulation : les données originales sont chiffrées puis encapsulées [15].

Le paquet du protocole ESP peut être représenté par le schéma suivant :

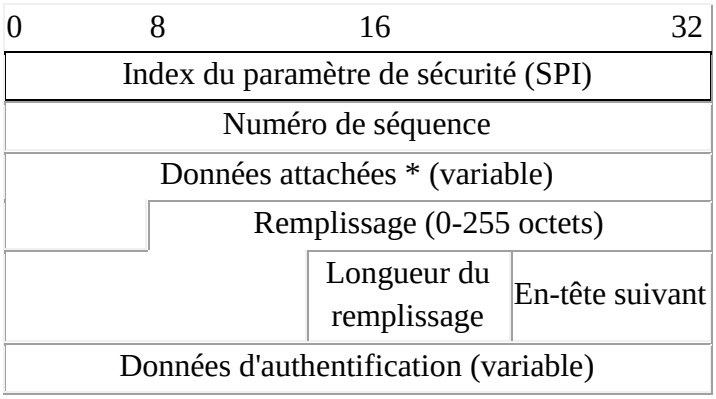

Figure 22 : Format du paquet ESP

Description des champs du paquet ESP :

- Index du paramètre de sécurité (SPI) : identifie les paramètres de sécurité en fonction de l'adresse IP ;
- Numéro de séquence : un compteur qui évite les attaques par répétition ;
- Données attachées : les données à transférer ;

- Remplissage (Bourrage) : permet d'obtenir une taille de bloc compatible avec le chiffrement ;
- Longueur du remplissage : indication du nombre d'octets de bourrage ;
- Entête suivant : identifie le protocole utilisé pour le transfert ;
- Données d'authentification : contient les informations nécessaires pour authentifier le paquet. Ce champ contient la valeur MAC.

2.4.3.3 Association de sécurité

L'association de sécurité SA (Security association) est une connexion qui fournit des services de sécurité au trafic qu'elle transporte [9]. Il s'agit d'une structure de données servant à stocker l'ensemble des paramètres de sécurité associés à une communication donnée, tels que les algorithmes de chiffrement et la fonction de hachage pour une communication accordée entre deux entités.

Une SA est unidirectionnelle ; en conséquence, protéger les deux sens d'une communication classique requiert deux associations, une dans chaque sens. Les services de sécurité sont fournis par l'utilisation soit de AH soit de ESP. Si AH et ESP sont tous deux appliqués au trafic en question, deux SA sont créées ; on parle alors de paquet (bundelle) de SA [14].

Chaque association est identifiée de manière unique à l'aide d'un triplet composé de :

- L'adresse de destination des paquets ;
- l'identifiant du protocole de sécurité (AH ou ESP) ;
- Un index des parametres de sécurité (Security Parameter Index, SPI).

Pour gérer les associations de sécurité active, on utilise une base de données des associations de sécurité (Security Association Database, SAD). Elle contient tous les parametres relatifs à chaque SA et sera consultée pour savoir comment traiter chaque paquet reçu ou à émettre.

2.4.3.4 Gestion des clés « Protocole IKE »

Les SA contiennent tous les paramètres nécessaires de l'IPsec, notamment les clés utilisées. Une SA peut être configurée manuellement dans le cas d'une situation simple, mais la règle générale est d'utiliser un protocole spécifique qui permet la négociation dynamique des SA et notamment l'échange des clés de session.

Le protocole négociation des SA développé par IPsec s'appelle «protocole de gestion des clefs et des associations de sécurité pour internet » (Internet Security Association and key Management Protocol, ISAKMP) [25]. ISAKMP est en fait inutilisable seul : c'est un cadre générique qui permet l'utilisation de plusieurs protocoles d'échange de clef et qui peut être utilisé pour d'autres mécanismes de sécurité que ceux de IPsec. Dans le cadre de standardisation de IPsec, ISAKMP est associé à une partie des protocoles SKEME et Oakley pour donner un protocole final du nom d'IKE (Internet Key Exchange).

Une négociation IKE pour l'établissement d'associations de sécurité se déroule en deux phases (Figure 23).

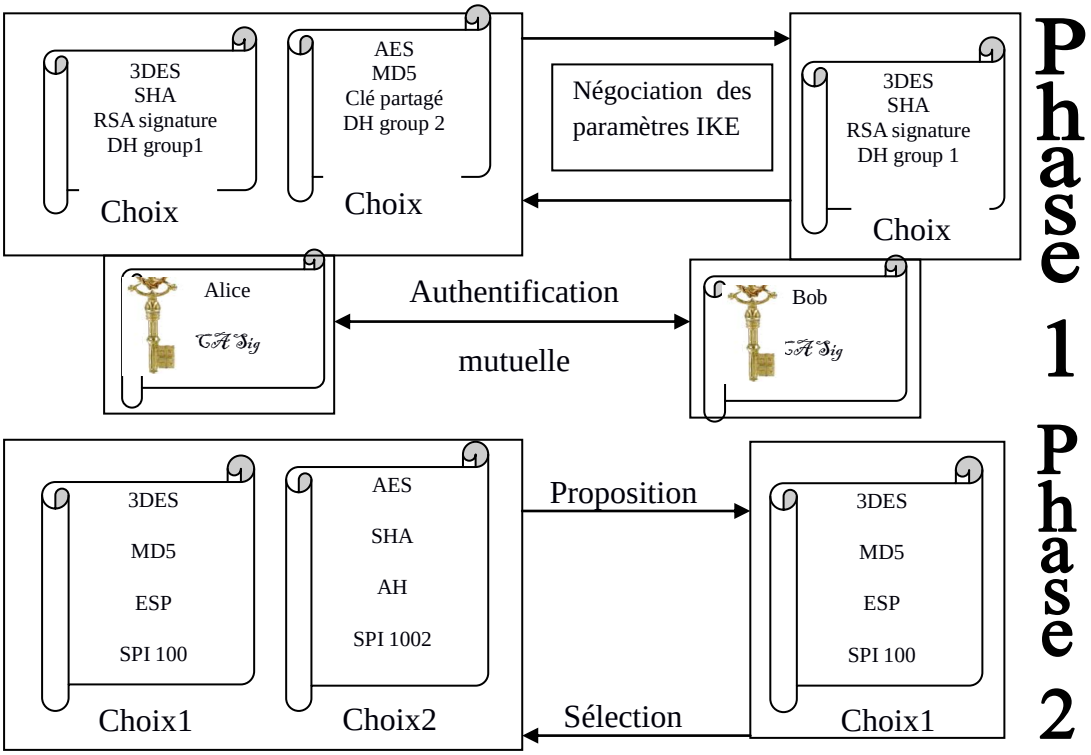

Figure 23 : Négociation d'une SA et des paramètres IPsec

La première phase est composée de six messages entre l'émetteur et le récepteur [9]:

- Les deux premiers messages servent à négocier les paramètres IKE : algorithme de chiffrement, fonction de hachage, méthode d'authentification et group pour DIFFIE-HELLMAN .
- Les deux seconds messages permettent l'établissement d'un secret partagé via l'utilisation de l'échange de valeurs publiques DIFFIE-HELLMAN .
- Les deux derniers messages sont chiffrés pour authentifier les deux entités.

Dans la seconde phase, les messages échangés durant cette phase sont protégés en authenticité et en confidentialité grâce aux éléments négociés durant la première phase. La phase 2 négocie à nouveau des paramètres de sécurité (ESP ou AH, algorithme de chiffrement, algorithme d'authentification, les cléfs).

2.4.3.5 Modes de fonctionnement

Pour chacun des mécanismes de sécurité d'IPsec, il existe deux modes : le mode transport et le mode tunnel.

2.4.3.5.1Mode transport

Ce mode est utilisé pour créer une communication entre deux hôtes qui supportent IPSec. Une SA est établie entre les deux hôtes. Les entêtes IP ne sont pas modifiées et les protocoles AH et ESP sont intégrés entre cette entête et l'entête du protocole transporté (Figure 24). Ce mode est souvent utiliser pour sécuriser une connexion Point-To-Point.

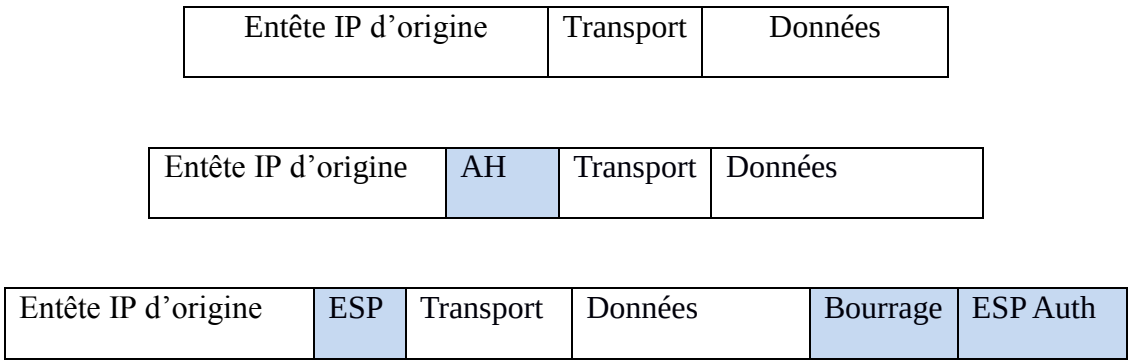

Figure 24 : Position AH et ESP en mode transport (IPV4)

2.4.3.5.2Mode tunnel

Ce mode est utilisé pour encapsuler les datagrammes IP dans IPsec. La SA est appliquée sur un tunnel IP. Ainsi, les entêtes IP originales ne sont pas modifiés et un entête propre à IPsec est ajouté (Figure 25). Le mode tunnel est donc utilisable à la fois sur des équipements terminaux et sur des passerelles de sécurité. Ce mode permet d'assurer une protection plus importante contre l'analyse du trafic, car il masque les adresses de l'expéditeur et du destinataire final [9].

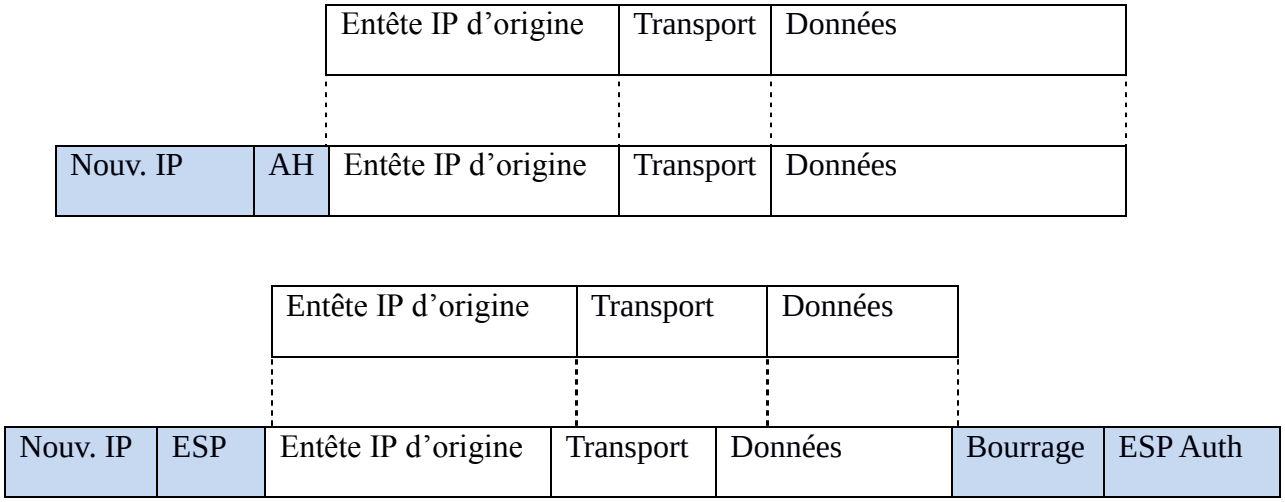

Figure 25 : Position AH et ESP en mode tunnel (IPV4)

2.4.3.6 Principe de fonctionnement

La figure suivante (Figure 26) montre les éléments du protocole IPsec, leurs positions et leurs interactions.

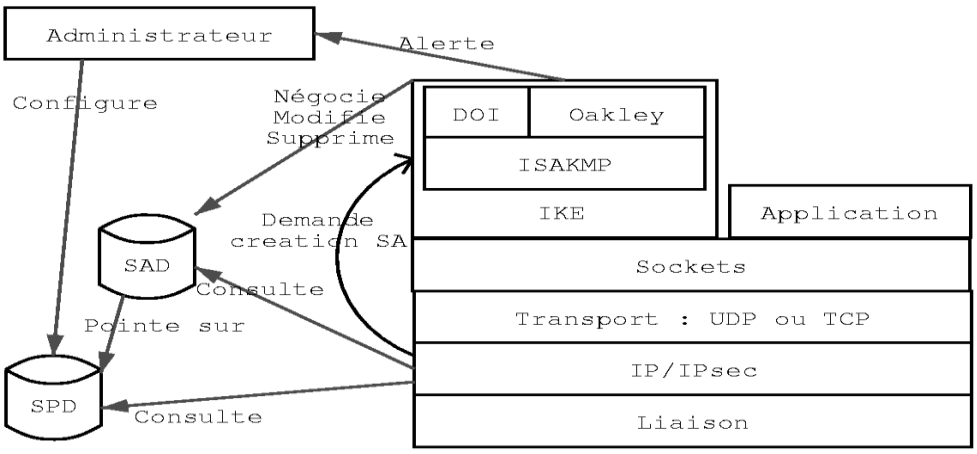

Figure 26 : Principe de fonctionnement de IPsec

On distingue deux situations :

- Trafic sortant : Lorsque la couche IPSEC reçoit des données à envoyer, elle commence par consulter la base de données des politiques de sécurité (SPD) pour savoir comment traiter ces données. Si cette base lui indique que le trafic doit se voir appliquer des mécanismes de sécurité, elle récupère les caractéristiques requises pour la SA correspondante et va consulter la base des SA (SAD). Si la SA nécessaire existe déjà, elle est utilisée pour traiter le trafic en question. Dans le cas contraire, IPSEC fait appel à IKE pour établir une nouvelle SA avec les caractéristiques requises.
- Trafic entrant : Lorsque la couche IPSEC reçoit un paquet en provenance du réseau, elle examine l'en-tête pour savoir si ce paquet s'est vu appliquer un ou plusieurs services IPSEC et si oui, quelles sont les références de la SA. Elle consulte alors la SAD pour connaître les paramètres à utiliser pour la vérification et/ou le déchiffrement du paquet. Une fois le paquet vérifié et/ou déchiffré, la Spd est consultée pour savoir si l'association de sécurité appliquée au paquet correspondait bien à celle requise par les politiques de sécurité.

Dans le cas où le paquet reçu est un paquet IP classique, la SPD permet de savoir s'il a néanmoins le droit de passer. Par exemple, les paquets IKE sont une exception. Ils sont traités par IKE, qui peut envoyer des alertes administratives en cas de tentative de connexion infructueuse.

2.4.3.7 Avantages

Le fait qu'IPSec soit un protocole de sécurité au niveau réseau, il permet d'avoir un niveau de sécurité très élevé par rapport aux autres solutions de couches applicatives [25]. Enumèrent les avantages d'IPSec et les présentes comme suit :

- Protection contre le contournement et l'analyse du trafic;
- Protocole transparent aux applications ;
- Mise en œuvre d'un VPN.

2.4.3.8 Inconvénients

Les problèmes rencontrés avec le protocole IPSec sont relatifs à l'ambiguïté documentaire, l'implémentation et à la redondance des fonctionnalités [25].

- Interopérabilité entre les implémentations;
- Redondances des fonctionnalités ;
- Broadcast et multicast;
- Aucune NATs.

2.5 Conclusion

Un système informatique peut être comparé à une maison. Il y a l'intérieur et l'extérieur. L'intérieur doit être agréable à vivre on doit s'y sentir en sécurité. L'intérieur est maintenu propre et en bon état, des équipements protègent des dangers potentiels (disjoncteur, robinet arrivée d'eau), nous prévoyons les petites catastrophes (bougies pour les coupures de courant, double des clés). L'extérieur est ressenti comme un agresseur potentiel, nous nous en protégeons, une porte blindée contre les effractions, les parois et ouvertures sont traitées pour éviter les déperditions thermiques et le bruit, nous choisissons qui nous faisons entrer.

Dans ce chapitre nous avons décrit différentes mesures de la sécurité informatique, ainsi que quelques protocoles de sécurité tels que SSH, SSL et IPsec.

- Le SSH est un outil très pratique pour gérer et administrer les serveurs Linux à distance dans un environnement contenant de multiples serveurs. Bien que le SSH chiffre les échanges pour les sécuriser, nous pouvons ajouter des sécurités supplémentaires aux accès SSH afin de rendre les tentatives d'attaques plus complexes.
- SSL propose une solution complète pour la sécurité. Mais permet de sécuriser uniquement les trafics Web et accepte seulement le protocole TCP au niveau transport [9].
- IPsec est un protocole au-dessus d'IP ; il support la sécurité à la fois du protocole UDP et du protocole TCP [9]

Chapitre 3 : Optimisation des règles de sécurités par l'utilisation des systèmes de détection d'intrusion

3.1 Introduction

La sécurité du système d'information de l'entreprise est un gros problème aujourd'hui. Malheureusement, les hackers et les intrus ont fait de nombreuses tentatives réussies, sur des réseaux des entreprises de haut niveau. Malgré les nombreuses méthodes ont été développées pour sécuriser l'infrastructure de réseau et de communication sur Internet, dont l'utilisation de pare-feu, le cryptage et les réseaux privés virtuels. Pour compléter cette politique de sécurité, il faut avoir des outils de surveillance pour auditer le système d'information et détecter d'éventuelles intrusions.

Une intrusion signifie la pénétration des systèmes d'information mais aussi la tentative des utilisateurs locaux d'accéder à de plus hauts privilèges que ceux qui leur sont attribués, ou tentatives des administrateurs d'abuser de leurs privilèges.

Un système de détection d'intrusion (ou IDS : Intrusion Detection System) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte) [26]. Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions.

Les premiers IDS ont été initiés par l'armée américaine, puis repris par les entreprises et intégrés par des projets open-source comme snort, Prelude [27],…

3.2 Différents types d'IDS

Un IDS a pour fonction d'analyser en temps réel ou différé les évènements en provenance des différents systèmes, de détecter et de prévenir en cas d'attaque [27]. Les buts sont nombreux :

- collecter des informations sur les intrusions ;
- gestion centralisée des alertes ;
- effectuer un premier diagnostic sur la nature de l'attaque permettant une réponse rapide et efficace.

Les systèmes de détection d'intrusion ou IDS peuvent se classer selon trois catégories majeures selon qu'ils s'attachent à surveiller :

- l'activité des machines : on parle d'IDS Système ou HIDS (Host based IDS)
- le trafic réseau : on parle d'IDS réseau ou NIDS (Network based IDS)
- IDS hybride rassemble les caractéristiques de plusieurs IDS

3.2.1 IDS coté hôte (HIDS)

Les IDS système (Host-based IDS) se situent directement sur une ressource à protéger [27]. Leur champ d'action pour détecter les intrusions se limite par conséquent à cette ressource (Figure 27). Toutefois, le type d'attaque concerne les différents éléments présents sur la même machine : appels système, processus en cours d'exécution, événements liés au système de fichiers, interactions avec l'utilisateur, etc.

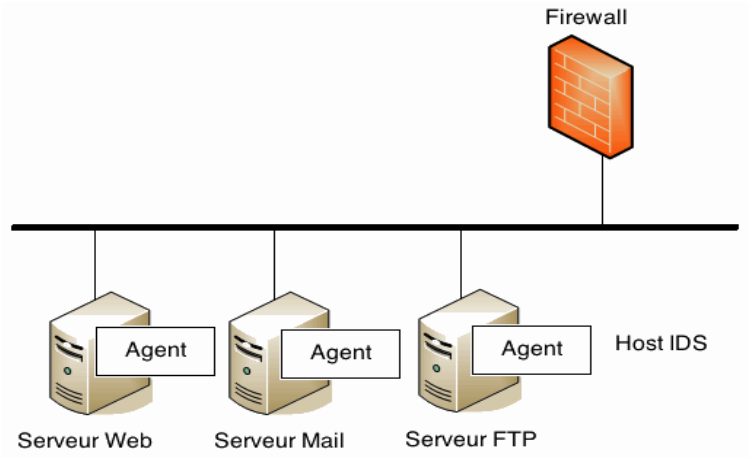

Figure 27 : Host Based IDS (HIDS)

3.2.2 IDS coté réseau (NIDS)

NIDS (Network IDS) permet d'analyser et d'interprété des paquets circulant sur le réseau (Figure 28). L'implantation d'un NIDS se fait par l'intermédiaire du placement de capteurs aux niveaux les plus stratégiques du réseau. Ces capteurs génèrent des alertes s'ils détectent des attaques et envoie les rapports à une console qui les analyse. Les capteurs et la console sont généralement sur un réseau distinct de celui de l'entreprise. Une carte réseau des capteurs est en mode «promisuous», de cette façon, les capteurs deviennent furtifs sur le réseau. L'autre carte réseau est connectée à la console. Les capteurs peuvent être placés avant ou après le pare-feu. Avant le pare-feu, il peut détecter les attaques qui seraient dirigés contre le pare-feu, et après le pare-feu il peut détecter les attaques que le pare feu aurait laissé passer. Les NIDS permettent donc de détecter les activités frauduleuses sur le réseau tout en étant bien sécurisés (puisqu'ils sont en mode furtif). A l'inverse des HIDS, les NIDS ne voient pas les impacts d'une attaque. Ils peuvent ne pas remarquer certains comportements anormaux, et doivent fonctionner en mode crypté ce qui ne facilite pas les analyses de trames [18].

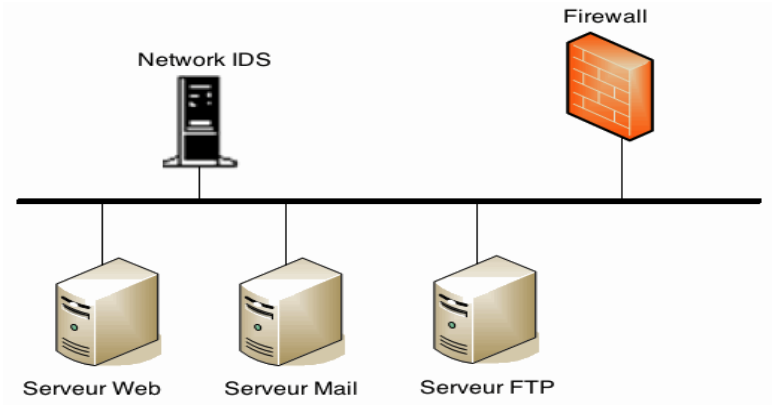

Figure 28 : Network IDS

3.2.3 IDS hybride

Les IDS Hybrides permettent de réunir les informations de diverses sondes placées sur le réseau. Leur appellation «hybride» provient du fait qu'ils sont capables de réunir aussi bien des informations provenant d'un système HIDS que d'un NIDS (Figure 29).

L'exemple le plus connu dans le monde Open-Source est Prelude. Ce Framework permet de stocker dans une base de données des alertes provenant de différents systèmes relativement variés [27].

Utilisant Snort comme NIDS, et d'autres logiciels tels que Samhain en tant que HIDS [21], il permet de combiner des outils puissants tous ensemble pour permettre une visualisation centralisée des attaques.

Les avantages des IDS hybrides sont multiples :

- Moins de faux positifs : certaines intrusions peuvent ne pas être détectées ;
- Meilleure corrélation ;
- Possibilité de réaction sur les analyseurs.

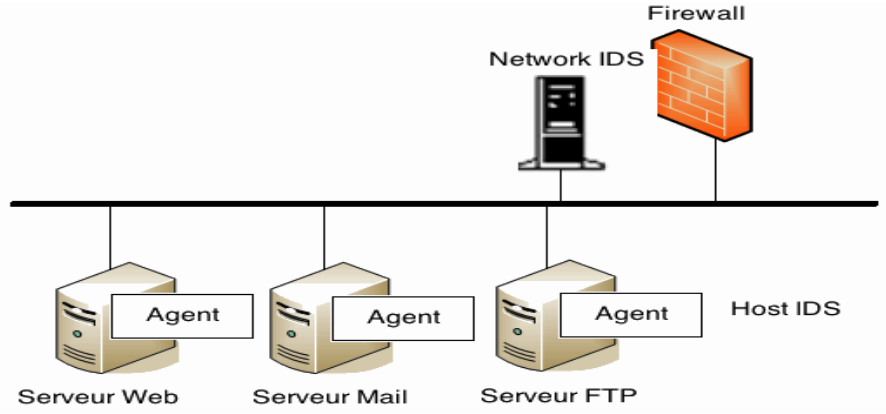

Figure 29 : Hybride IDS

3.3 Mécanismes de détection

Les IDS disposent de deux mécanismes principaux, pour détecter les intrusions. La première consiste en la détection de signatures d'attaques sur un réseau répertoriées dans une base de connaissance. La deuxième permet de détecter une activité suspecte d'un utilisateur sur un poste donné [27].

3.3.1 Approche par scénario

La détection par scénario s'appuie sur une base de données de signatures de scénarios d'attaques connus (knowledge based detection)[20]. Par conséquent, le principe consiste à comparer le comportement à une attaque connue. Ce mécanisme de détection est efficace contre les attaques clairement répertoriées, mais pas contre de nouvelles attaques. Elle génère peu de faux-positifs, c'est-à-dire de fausses alertes, mais nécessite une description très précise des scénarios auxquels se comparer.

Les signatures sont principalement utilisées de deux manières différentes : un langage de haut niveau décrit les scénarios (expressions rationnelles, logique temporelle, etc.), ou on se concentre sur les changements d'états (automates, etc.)[21].

Trois familles de méthodes sont utilisées par les IDS à signature qui se basent tous sur la recherche d'un profil connu d'attaque.

3.3.1.1 Recherche de motifs (pattern matching)

La méthode la plus connue et la plus à facile à comprendre. Elle se base sur la recherche de motifs (chaînes de caractères ou suite d'octets) au sein du flux de données [20]. L'IDS comporte une base de signatures où chaque signature contient les protocoles et port utilisés par l'attaque ainsi que le motif qui permettra de reconnaître les paquets suspects [21]. Le principal inconvénient de cette méthode est que seules les attaques reconnues par les signatures seront détectées. Il est donc nécessaire de mettre à jour régulièrement la base de signatures. Un autre inconvénient est que les motifs sont en général fixes. Or une attaque n'est pas toujours identique à 100%. Le moindre octet

différent par rapport à la signature provoquera la non détection de l'attaque. Pour les IDS utilisant cette méthode, il est nécessaire d'adapter la base de signatures en fonction du système à protéger. Cela permet non seulement de diminuer les ressources nécessaires et donc augmenter les performances ; mais également réduire considérablement le nombre de fausses alertes et donc faciliter le travail des administrateurs réseaux qui analyseront les fichiers d'alertes. Cette technique est également utilisée dans les anti-virus.

3.3.1.2 Recherche de motifs dynamiques

Le principe de cette méthode est le même que précédemment mais les signatures des attaques évoluent dynamiquement. L'IDS est de ce fait doté de fonctionnalités d'adaptation et d'apprentissage.

3.3.1.3 Analyse heuristique et détection d'anomalies

Le but de cette méthode est, par une analyse intelligente, de détecter une activité suspecte out outre autre anomalie.

Par exemple : une analyse heuristique permet de générer une alarme quand le nombre de sessions à destination d'un port donné dépasse un seuil dans un intervalle de temps prédéfini.

3.3.2 Approche comportementale

Le but d'une détection par approche comportementale (anomaly detection), permet de signaler toutes anomalies sur un Système d'information. En effet il passe par une première phase dite d'apprentissage du comportement normal du système. Tout comportement suspect sera déclaré et se fera en fonction des modèles ou profils établis lors de la première phase d'un comportement normal. Plusieurs approches peuvent être utilisées pour la méthode de détection comportementale [19] :

- **Probabiliste** : la construction des profils se font sur la définition d'un fonctionnement des applicatifs. Une alarme est levée en fonction des évènements observés à savoir l'établissement des règles ou l'apprentissage des probabilités liées à chaque séquence d'évènements [17].
- **Statistique** : la construction des profils se font à partir d'une quantification de certains paramètres liés à un évènement (Taux d'occupation mémoire, charge CPU…). Une alarme est levée en fonction du taux de déviation du comportement normal et celui observé [17].
- **Réseaux de neurones** : La construction des profils se font via une reconnaissance d'une suite d'opérations effectuées par l'utilisateur (outils de travail, activités enregistrées). Le but étant de prédire la prochaine action de l'utilisateur, une alarme sera levée en cas d'échec [17].

3.4 Quelques systèmes de détection d'intrusions

3.4.1 Critères de choix d'un IDS

Actuellement les systèmes de détection d'intrusion sont réellement devenus indispensables. Ils s'intègrent donc toujours dans un contexte et une architecture qui imposent des contraintes pouvant être très diverses [21]. C'est pourquoi il n'existe pas de grille d'évaluation unique pour ce type d'outil. Pourtant un certain nombre de critères peuvent être relevés.

- **Fiabilité** : Un détecteur d'intrusion doit être fiable ; les alertes qu'il génère doivent être justifiées et aucune intrusion ne doit pouvoir lui échapper.
- **Réactivité** : Un IDS doit être capable de détecter les nouveaux types d'attaque le plus rapidement possible ; pour cela il doit rester constamment à jour. Des capacités de mise à jour automatique sont pour ainsi dire indispensables.

- **Facilité de mise en œuvre et adaptabilité** : Un IDS doit être facile à mettre en œuvre et doit pouvoir surtout s'adapter au contexte (matériels, etc. . .) dans lequel il doit opérer.
- **Performances** : la mise en place d'un IDS ne doit en aucun cas affecter les performances des systèmes surveillés. De plus, il faut toujours avoir la certitude que l'IDS a la capacité de traiter toute l'information à sa disposition.
- **Multicanal** : Un bon IDS doit pouvoir utiliser plusieurs canaux d'alerte (email, pager, téléphone, fax...) afin de pouvoir garantir que les alertes seront effectivement émises. L'IDS doit donner un maximum d'information sur l'attaque détectée afin de préparer la réaction.
- **Classification** : il doit être aisé de hiérarchiser la gravité des attaques détectées afin d'adapter le mode d'alerte.

3.4.2 Quelques outils d'IDS

Voici quelques outils qui sont disponibles, on les distingue selon leur méthode de détection ainsi que sur leur modèle économique [24].

Nom de l'IDS	HIDS	NIDS	Comportementale	Scénario	Payant	Libre
Attack Mitigator		X		X	X	
Bro		X		X		X
Cisco IPS				X	X	
Dragon		X		X	X	
Prelude-IDS		X		X		X
SNORT		X	X	X	X	X

Table 3 : Table des caractéristiques de quelques IDS

Dans ce travaille, nous avons opté pour le IDS SNORT pour les raison suivants :

- Actuellement, c'est le logiciel le plus répandu avec plus de 2 000 000 téléchargements.
- Il bénéficie d'une mise à jour en temps réel (une version payante : OINKMAster via SourceFile et une autre gratuite : Bleeding via CERT), d'où la grande réactivité.
- Il propose une architecture modulaire composée de :
    - Préprocesseurs ;
    - Moteur de détection ;
    - Système d'alerte et d'enregistrement de log ;
    - Modules de sortie (pour enregistrer les logs dans des bases de données par exemple) ;
    - Décodeur de paquet.

3.5 Système de détection d'intrusion en pratique : Snort

3.5.1 Définition

Snort est un système de détection d'intrusion libre sous licence GPL. À l'origine écrit par Martin Roesch, il appartient actuellement à Sourcefire. Des versions commerciales intégrant du matériel et des services de supports sont vendues par Sourcefire.

Snort est capable d'effectuer aussi en temps réel des analyses de trafic et de logger les paquets sur un réseau IP. Il peut effectuer des analyses de protocole, recherche/correspondance de contenu et peut être utilisé pour détecter une grande variété d'attaques et de sondes comme des dépassements de buffers, scans, attaques sur des CGI, sondes SMB, essai d'OS fingerprintings et bien plus. Cependant, comme tout logiciel, Snort n'est pas infaillible et demande une mise à jour régulière.

Snort peut également être utilisé avec d'autres projets open sources tels que SnortSnarf, ACID, sguil et BASE (Basic Analysis and Security Engine) afin de fournir une représentation visuelle des données concernant les éventuelles intrusions [22].

3.5.2 Architecture de Snort

Lorsqu'un comportement suspect est intercepté par Snort en fonction des signatures, Snort inscrit les événements. Il est possible d'inscrire les logs en base de données directement. Néanmoins, dans un souci d'optimisation (libération de ressources), nous utiliserons Barnyard, une couche d'abstraction de Snort. Ainsi, Snort inscrira les événements dans des logs au format unifié (Fast Unified Logging) et ces derniers seront exploités par Barnyard pour une inscription en base de données [23]. Guardian assurera le blocage des adresses IP détectées comme suspectes. Par ailleurs, BASE et/ou SGUIL permettront d'analyser ces attaques (Figure 30).

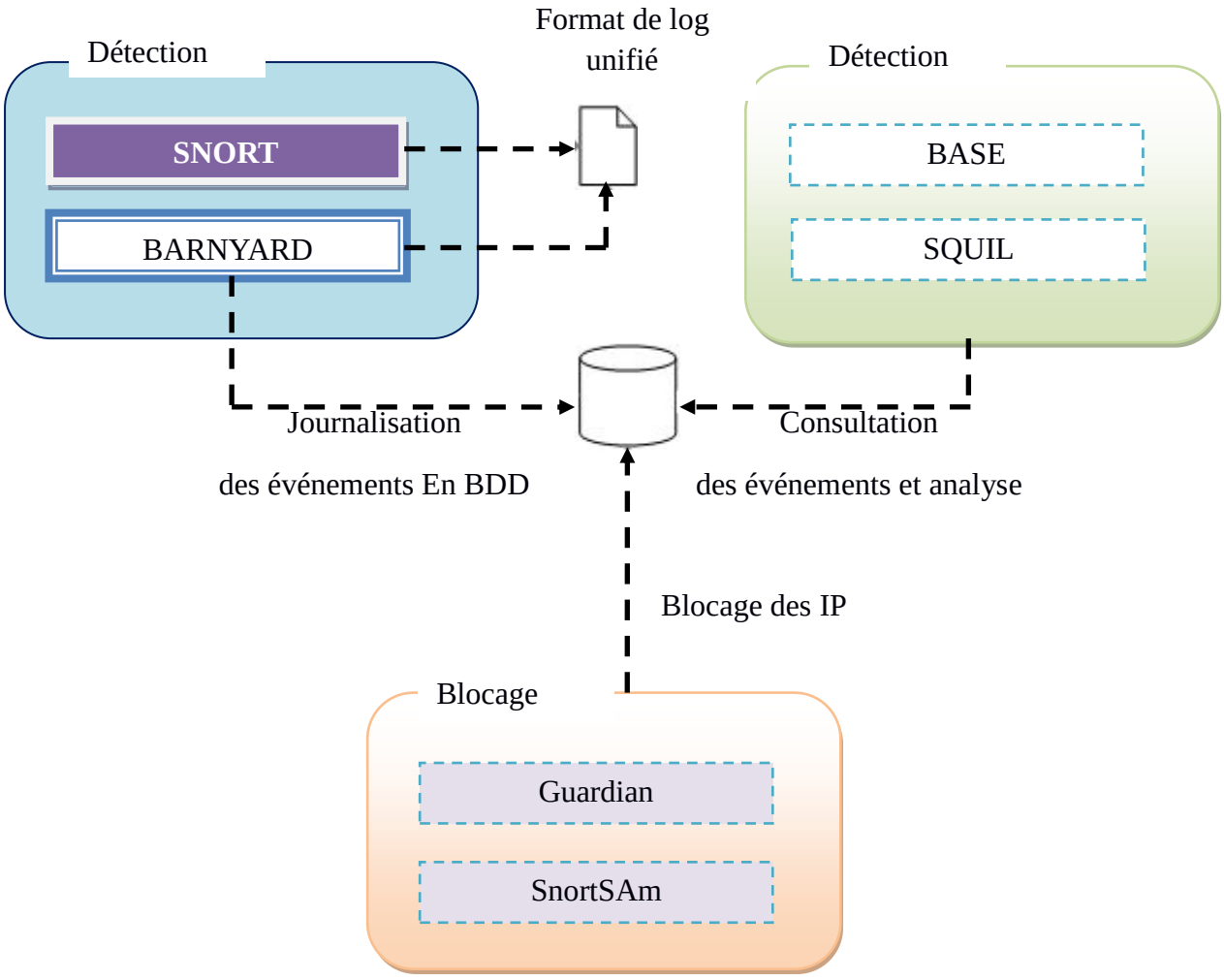

Figure 30 : Architeture de snort

3.5.3 Positionnement de SNORT

L'emplacement physique de la sonde SNORT sur le réseau a un impact considérable sur son efficacité [23]. Dans le cas d'une architecture classique, composée d'un Firewall et d'une zone DMZ. Trois positions sont généralement envisageables (Figure 31) :

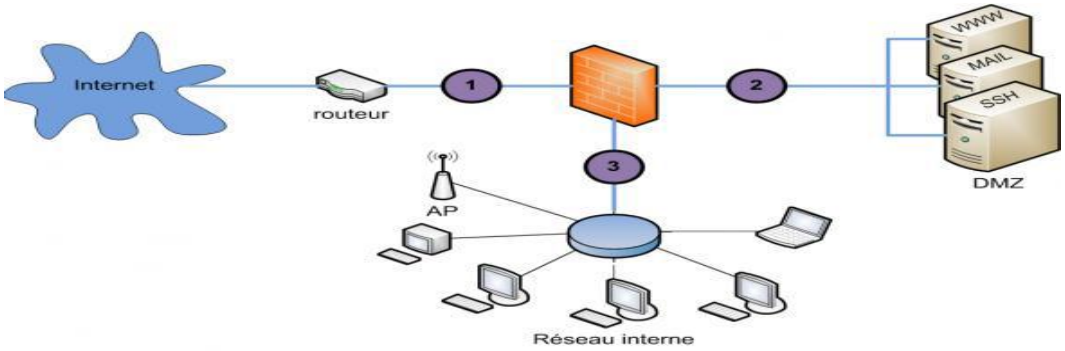

Figure 31 : Positionnement de Snort

- **Position 1** : Sur cette position, Snort va pouvoir détecter l'ensemble des attaques frontales, provenant de l'extérieur, en amont du firewall. Ainsi, beaucoup d'alertes seront remontées ce qui rendra les logs difficilement consultables.
- **Position 2** : Si Snort est placé sur la DMZ, il détectera les attaques qui n'ont pas été filtrées par le firewall et qui relèvent d'un certain niveau de compétence. Les logs seront ici plus clairs à consulter puisque les attaques bénignes ne seront pas recensées.
- **Position 3** : Snort peut ici rendre compte des attaques internes, provenant du réseau local de l'entreprise. Il peut être judicieux d'en placer un à cet endroit étant donné le fait que 80% des attaques proviennent de l'intérieur. De plus, si des trojans ont contaminé le parc informatique (navigation peu méfiante sur internet) ils pourront être ici facilement identifiés pour être ensuite éradiqués.

Dans ce travaille je m'intéresse à la détection des intrusions qui proviennent de l'extérieur, pour les bloquer au niveau de Firewall.

3.5.4 Modes de fonctionnement

Il existe 4 modes d'exécutions de Snort [23] :

- **Mode sniffer** : C'est un snif de réseau classique. Inutile de s'y attarder, d'autres logiciels comme wireshark le font très bien, et la valeur réelle de Snort n'est pas là.
- **Mode Packet logger** : De même que le mode sniffer, sauf qu'il écrit le résultat de son observation dans un fichier log.
- **Mode NIDS (Network Intrusion Detection System)** : Cela devient plus intéressant. Ce mode fait l'objet de mon travail. Il s'agit de l'utilisation de Snort avec analyse du trafic aux vues de règles de sécurités actualisées.
- **Mode IPS (IPS= Intrusion Prevention System) ou Snort inline** : Le mode IPS n'est plus Snort à proprement parler. Il s'agit d'une autre version basée sur Snort 2.6 appelée Snort inline. Cette version permet de modifier ou de rejeter des paquets.

3.5.5 Règles de SNORT

Les règles de SNORT sont composées de deux parties distinctes : le header et les options (Figure 31). Le header permet de spécifier le type d'alerte à générer (alert, log et pass) et d'indiquer les champs de base nécessaires au filtrage : le protocole ainsi que les adresses IP et ports sources et destination. Les options, spécifiées entre parenthèses, permettent d'affiner l'analyse, en décomposant la signature en différentes valeurs à observer parmi certains champs du header ou parmi les données.

action	protocole	adress1	port1	direction	adresse2	port2	Options (msg, content ….etc)

Figure 32 : Règle de Snort

Champ action : alert, log, pass
Champ protocole : tcp, udp, icmp
Champs adresses source et destination : src, dest, any
Port src / dest : any, nb port, plage de ports avec p1:pn
Le champs direction : renseignent Snort sur la direction des échanges réseau (->, <->, <-).

La partie options est constituée de plusieurs champs qui assurent l'analyse du contenu des paquets réseau avec plus de finesse. Notons que la manipulation de ces champs nécessite une grande maîtrise des protocoles réseau pour pouvoir décrire les signatures des attaques à détecter.
Pour chaque option le format est nom option : valeur1 [, valeur2,…] ci-dessous quelques options utilisées dans la création des règles.

msg: spécifie le message qui sera affiché dans le log et dans l'alerte
reference: fait référence aux sites expliquant l'attaque détectée (bugtraq , CVE, …etc.)
classtype: définit la classe de l'attaque (troyen, shellcode …etc)
ttl: spécifie la valeur du TTL du paquet
flags: spécifie la présence d'un flag TCP dans le paquet (SYN, Fin, …etc) ..etc

Exemple: alert icmp any any -> $HOME_NET any (msg: »ICMP test »; sid:10000001;)

3.6 Outils de reporting web

3.6.1 BASE (Basic Analysis and Security Engine)

Basic Analysis and Security Engine est une interface web sous licence GPL écrit en php et bâti sur le code d'A.C.I.D [23], qui permet de visualiser les alertes générées par le logiciel de détection d'intrusion Snort. L'interface permet le classement des alertes en groupe, l'affichage de diagrammes et la recherche des alertes selon différents critères (Figure 33).

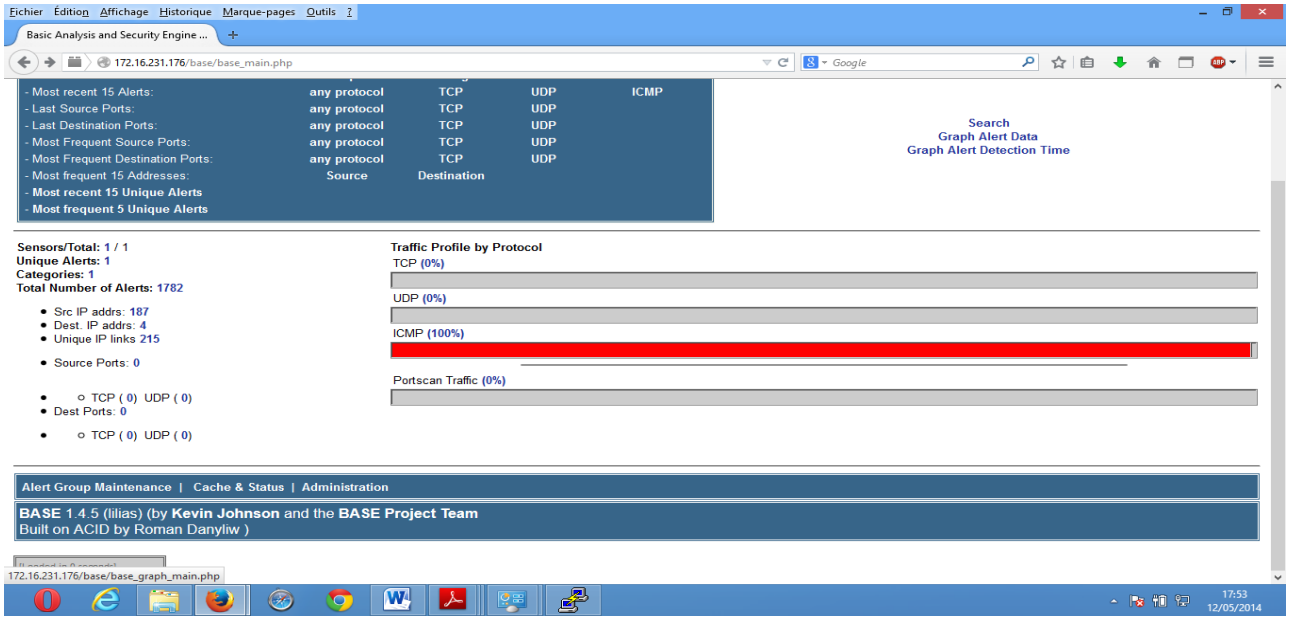

Figure 33 : Capture d'écran BASE

3.6.2 SnortReport

SnortReport est une interface graphique permet d'affiche et d'analyser les alertes générées par le système de détection d'intrusion Snort (Figure 34, Figure 35), stockées dans une base de données MySql.

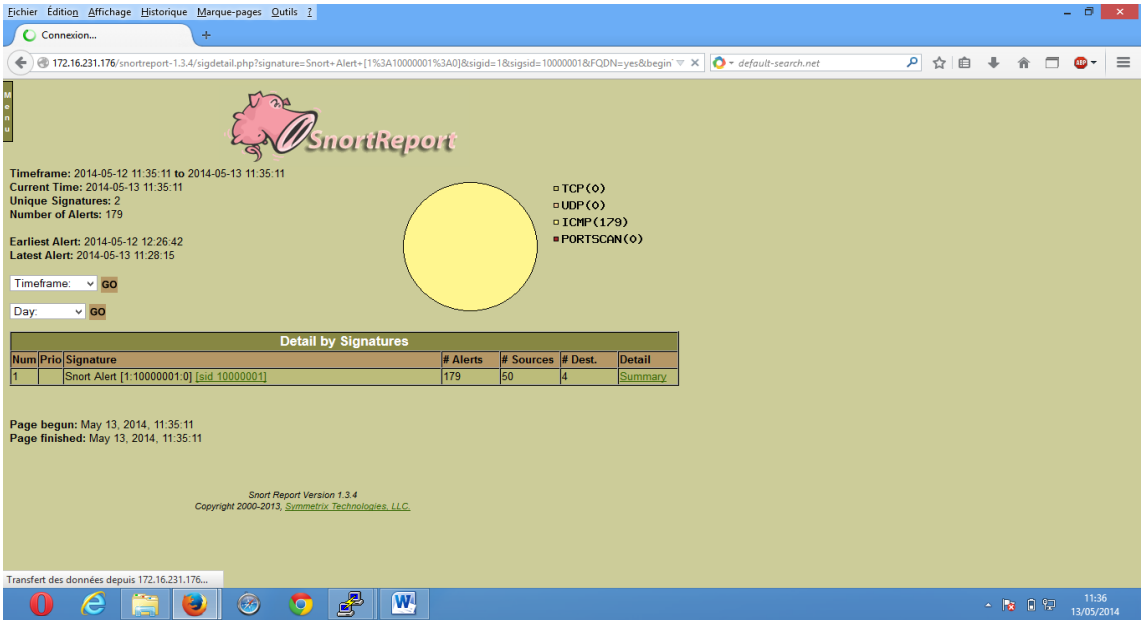

Figure 34 : Liste des alertes afficher par SnortReport

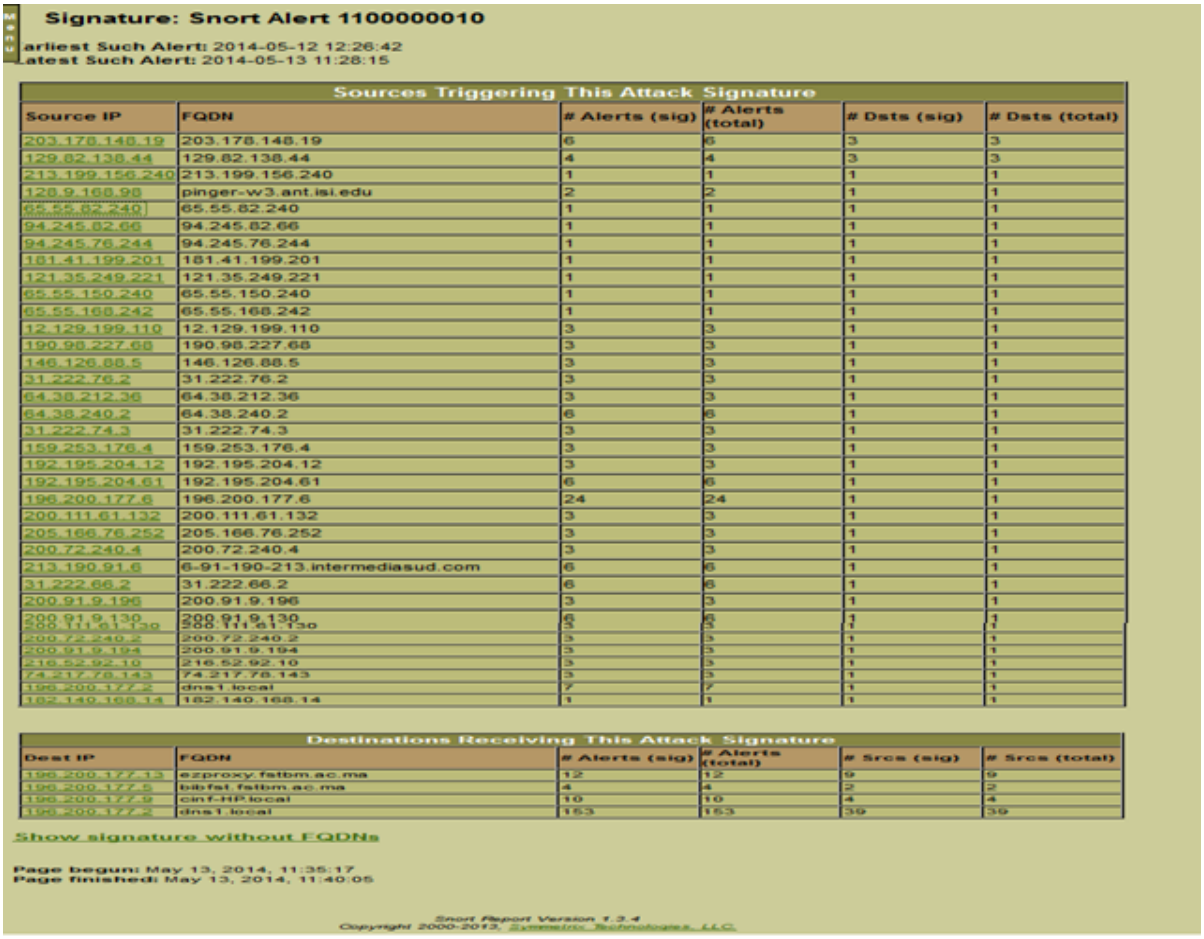

Figure 35 : Détails des alertes afficher par SnortReport

3.7 Conclusion

Après s'être intéressé aux problèmes de sécurité sur Internet, et compte tenu de la multitude des attaques sur Internet, nous avons vu qu'il était préférable de disposer d'un détecteur d'intrusion réseau. Ainsi, après avoir installé Snort et l'interface web BASE e SnortReport pour détecter, alerter et analyser ces intrusions, nous avons analysé en particulier l'une de ces alertes. Cependant, si l'on a mis en place une sonde, cela ne permet pas pour autant de sécuriser le réseau, mais aide à mieux en percevoir ses défauts, quitte à améliorer par la suite les règles de sécurité des pare-feu et à utiliser davantage d'outils et technologies de sécurité réseau. De plus un tel outil nous permet d'obtenir des statistiques sur les tentatives d'intrusion, réussies ou non, et de les tracer afin d'obtenir des pistes de prévention, voire de répression.

Snort, BASE et SnortReport sont des logiciels très connus, davantage d'informations sont disponibles sur Internet.

Chapitre 4 : Architecture de l'infrastructure virtuelle

4.1 Introduction

La virtualisation est l'ensemble des technologies matérielles et/ou logiciels qui permettent de faire fonctionner sur une seule machine physique plusieurs systèmes d'exploitation et/ou plusieurs applications, séparément les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes [30] [31].

Les débuts de la virtualisation remontent au milieu des années 60 sur la plate-forme de superordinateurs mainframe d'IBM. À cette époque, les machines virtuelles étaient appelées des pseudo-machines. À l'origine, l'ordinateur central utilisait le programme de contrôle pour allouer des ressources et isoler les différentes instances des pseudo-machines les unes des autres [31]. Les deux chercheurs en informatique, Gerald J. Popek et Robert P. Goldberg, en 1974, ont posé les bases de la virtualisation dans leur article «Formal Requirements for Virtualizable Third Generation Architectures», publié dans Communications of the ACM (Association for Computing Machinery) en juillet 1974 (Hess, et al., 2010)[31]. À la fin des années 90, l'équipe de Mendel Rosemblum met au point un procédé permettant de transposer le concept « IBM » vers les processeurs x86 [31]. Rosemblum fonde en 1999 la société VMware et lance le produit VMware Workstation 1.0. L'hyperviseur ESX sort, quant à lui, en 2001. Il faut attendre 2008 pour que Microsoft lance son hyperviseur Hyper-V pour tenter de contrer la position dominante de VMware sur le marché de la virtualisation [31].

Du côté des logiciels libres de virtualisation, la première version publique de Xen est disponible en 2003 [32]. En 2007, l'éditeur Citrix Systems rachète Xen source, et annonce fin 2009 que sa surcouche propriétaire serait Open Source [32]. Citrix reverse le code développé dans le projet communautaire Xen Cloud Platform (XCP) dont la première version stable est sortie en mars 2011 [32]. Deux fonctionnalités avancées de l'API de Citrix ne sont pas libérées : la répartition automatique de charge et le failover automatique. Ces fonctionnalités sont disponibles dans l'édition payante de Xen. En 2013, Citrix XenServer est totalement Open Source [32].

Ainsi la virtualisation peut se définir comme un processus consistant à créer une version virtuelle d'une entité physique [30] [35]. Elle s'applique à des ordinateurs, à des systèmes, à des périphériques de stockages, à des applications ou à des réseaux [35]. La virtualisations des serveurs c'est la clé de ce processus. L'objectif de ce processus c'est la mutualisation des capacités de chaque serveur, de réaliser des économies sur le matériel [35] ce qui entraine une diminution de la consommation électrique et des gains d'efficience [37]. Nous voyons que la virtualisation repose sur trois éléments importants :

- L'abstraction des ressources informatiques ;
- La répartition des ressources par l'intermédiaire de différents outils, de manière à ce que celles-ci puissent être utilisées par plusieurs environnements virtuels ;
- La création d'environnements virtuels.

4.2 Domaines de la virtualisation

4.2.1 Virtualisation d'applications

La virtualisation d'application est une technologie logicielle qui va permettre d'améliorer la portabilité et la compatibilité des applications en les isolants du système d'exploitation sur lequel elles sont exécutées [34]. Elle consiste à encapsuler l'application et son contexte d'exécution système dans un environnement cloisonné [34]. La virtualisation d'application va nécessiter l'ajout

d'une couche logicielle supplémentaire entre un programme donné et le système d'exploitation ; son but est d'intercepter toutes les opérations d'accès ou de médication de fichiers ou de la base de registre afin de les rediriger de manière totalement transparente vers une localisation virtuelle (généralement un fichier). Puisque cette opération est transparente, l'application n'a pas notion de son état virtuel. Le terme virtualisation d'application est trompeur puisqu'il ne s'agit pas de virtualiser l'application mais plutôt le contexte au sein duquel elle s'exécute [34][36].

4.2.2 Virtualisation de réseaux

La virtualisation des réseaux consiste à partager une même infrastructure physique au profit de plusieurs réseaux virtuels isolés [42]. Un VLAN (Virtual Local Area Network) est un réseau local regroupant un ensemble de machines de façon logique (Figure 36) et non physique [42]. Puisqu'un VLAN est une entité logique, sa création et sa configuration sont réalisées de manière logicielle et non matérielle [42].

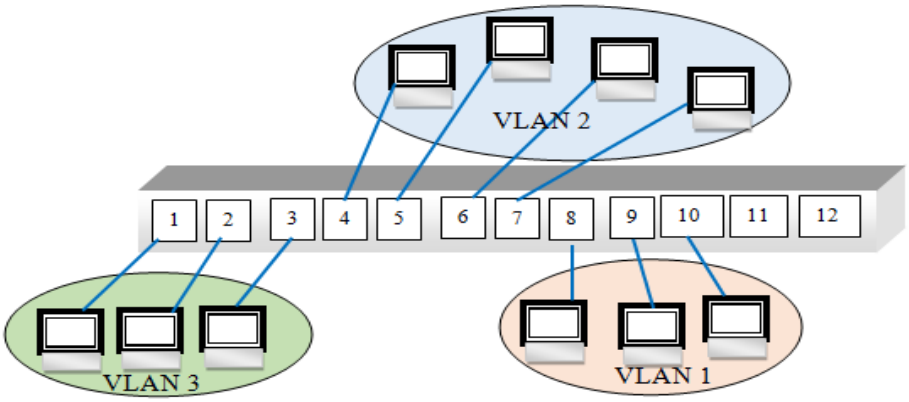

Figure 36 : Virtual Local Area Network (VLAN)

De plus, la virtualisation réseau permet de créer des entités réseaux sous forme logicielle et de les intégrer dans la couche hyperviseur (Figure 37), de sorte de les dissocier du matériel physique sous-jacent [40][41]. Un hyperviseur réseau virtualise toutes les couches réseau, de la couche 2 à la couche 7 du modèle OSI (switch, routage, ACL, Firewall, etc) en tant que software [43]. Ceci permet de créer un réseau virtuel isolé. Le réseau virtuel est indépendant du réseau physique et complètement agnostique [43]. Il vient s'interfacer au-dessus de la couche physique et ne demande aucun équipement supplémentaire.

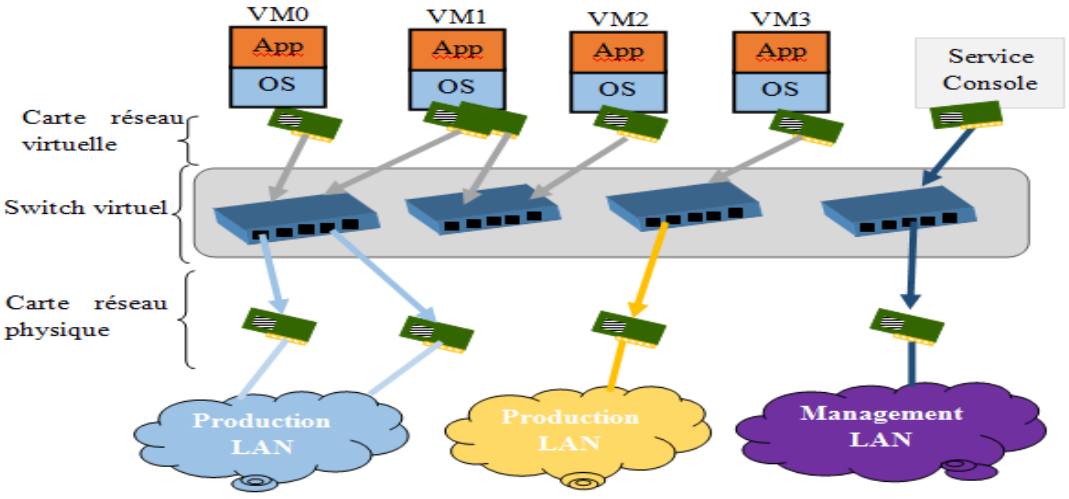

Figure 37 : Virtualisation réseau au niveau de hyperviseur

4.2.3 Virtualisation de stockage

La virtualisation du stockage est un procédé qui va séparer la présentation logique et la réalité physique de l'espace de stockage afin de limiter l'impact des modifications structurelles de l'architecture de stockage [44]. La couche de virtualisation présente un espace de stockage logique et s'occupe de faire le lien avec la localisation physique des données [45]. Les hyperviseurs assurent la liaison vers ces espaces de stockage, appelés volumes physiques [44][46].
Les VM peuvent accéder aux volumes physiques de différentes façons. Ces différentes méthodes d'accès sont présentées dans la figure suivante (Figure 38).

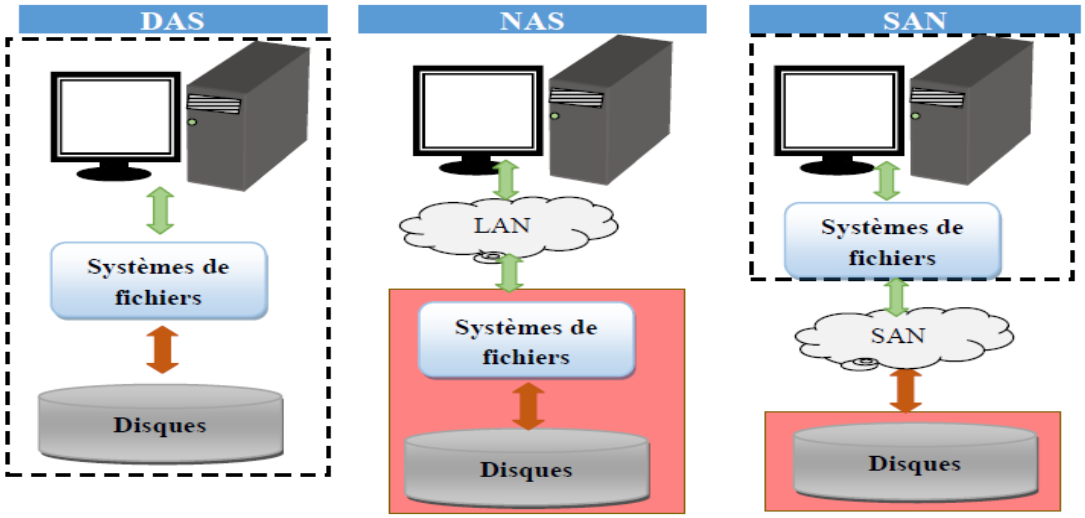

Figure 38 : Méthodes d'accès aux volumes de stockage : DAS, NAS, SAN

Les VM accèdent au stockage par trois méthodes [44][46]:

- Le Direct Attached Storage (DAS) qui est un stockage directement attaché au serveur hôte par le biais de technologies ATA, SATA, eSATA, SCSI, SAS et Fibre Channel5 (FC) via des câbles dédiés ;
- Le Storage Area Network (SAN), qui est un réseau de zone de stockage. Le chemin réseau, via les protocoles FC ou iSCSI6, est dédié aux échanges entre l'hyperviseur et la baie de stockage SAN ;
- Le Networked Attached Storage (NAS) qui est un stockage en réseau et qui apparait comme un serveur de fichiers distant via un réseau local ou étendu. Le NAS utilise des protocoles spécialisés d'accès aux fichiers et de partage de fichiers comme NFS, CIFS, SMB.

4.2.4 Virtualisation de serveurs

Le principe de la virtualisation des serveurs est simple. On considère le serveur comme un ensemble de ressources (CPU / RAM / DISQUES DUR / RESEAUX) que l'on allouera de manière statique ou dynamique aux serveurs virtuels [47]. Ces serveurs virtuels ne voient que les ressources qui leur sont allouées et sont donc isolés les uns des autres. Plusieurs systèmes d'exploitation (OS) peuvent donc coexister sur le même serveur physique [47] (Figure 39).

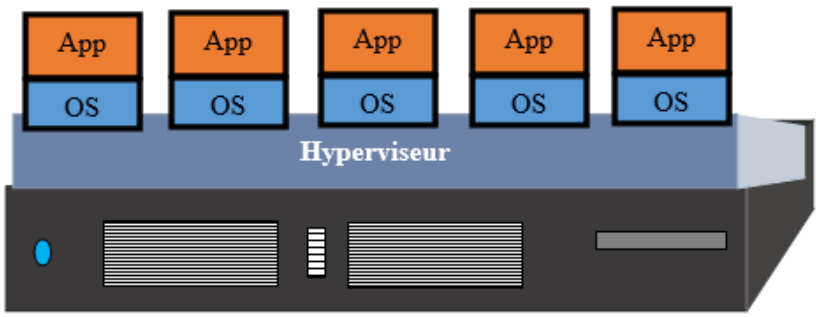

Figure 39 : Virtualisation des serveurs

D'un point de vue purement technique, il n'est pas simple de faire comprendre au système invité qu'il virtualisé. Un OS a pour habitude de discuter au plus bas niveau avec le matériel. Il existe plusieurs approches pour ce faire [47] :

- Faire croire au système virtualisé qu'il tourne au plus bas niveau, il faudra pour ce faire intercepter toutes les communications avec le matériel pour les modifier à la volée ce qui s'avère couteux en termes de performances.
- Modifier l'OS virtualisé afin qu'il ne communique plus au plus bas niveau.

La virtualisation des serveurs comportent quand même un lot conséquent d'avantages [47] :

- La maintenance matérielle centralisée, la consommation électrique diminuée, l'espace requis pour les serveurs est diminué, et les besoins en climatisation également ;
- Le rachat de matériel n'est plus obligatoire pour chaque déploiement ;
- Les erreurs d'administration ne sont plus fatales (Snapshots / Retour en arrière) ;
- La sauvegarde est moins complexe puisque les VM (Virtual Machines) sont en réalité de simples fichiers ;
- Les déploiements sont moins complexes ;
- La possibilité de créer des architectures hautement disponibles ;
- La gestion du changement est plus facile ;
- La supervision en temps réelle est plus simple également.

Enfin, on peut résume que La virtualisation des serveurs permet à l'entreprise d'optimiser l'utilisation de ses ressources serveur de façon à réduire le nombre de serveurs requis [48]. La consolidation des serveurs qui en résulte est synonyme de gain d'efficacité et de réduction des coûts [34]. La virtualisation de serveurs s'inscrit dans une tendance globale qui tend à promouvoir la virtualisation au sein des entreprises en faisant notamment appel à la virtualisation de stockage et à la virtualisation de réseaux.

4.3 Types de la virtualisation

4.3.1 Isolation

L'isolation (aussi appelée cloisonnement) est une technique qui intervient au sein d'un même système d'exploitation. Elle permet de séparer un système en plusieurs contextes ou environnements (Figure 40). Chacun d'entre eux est régi par l'OS hôte, mais les programmes de chaque contexte ne peuvent communiquer qu'avec les processus et les ressources associées à leur propre contexte [36].
L'isolation est utilisée sous Unix depuis longtemps pour protéger les systèmes [50]. Via des mécanismes comme chroot ou jail, il est possible d'exécuter des applications dans un environnement qui n'est pas celui du système hôte, mais un « mini système » ne contenant que ce dont l'application a besoin, et n'ayant que des accès limités aux ressources [36].

Avec l'isolation, l'espace noyau n'est pas différencié, il est unique, partagé entre les différents contextes [50]. Mais on définit de multiples espaces utilisateurs cloisonnés. C'est ainsi que l'on peut

faire cohabiter différentes distributions de système d'exploitation, à condition qu'elles partagent le même noyau [35][36].

L'isolation des contextes est une solution légère, tout particulièrement dans les environnements Linux [36]. L'unicité du noyau reste bien sûr une petite limitation. D'une part en termes de robustesse, puisqu'un plantage du noyau – fort heureusement très rare dans le monde Linux – plante simultanément tous les environnements. D'autre part dans les utilisations possibles, puisque typiquement ce mode ne conviendra pas pour valider une nouvelle version de noyau.

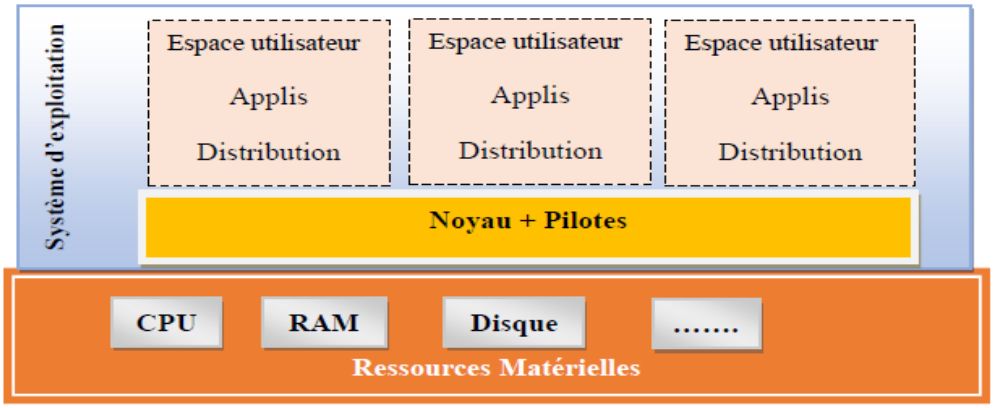

Figure 40 : Virtualisation par isolation

4.3.2 Virtualisation complète

La virtualisation complète, consiste à simuler un ordinateur complet, de façon que le système d'exploitation invité n'ait pas conscience d'être virtualisé [31]. Le système d'exploitation qui est virtualisé n'a aucun moyen de savoir qu'il partage le matériel avec d'autres OS [32][50]. Ainsi, l'ensemble des systèmes d'exploitation virtualisés s'exécutant sur une unique machine physique (Figure 41), peuvent fonctionner de manière totalement indépendante les uns des autres et être vu comme des machines à part entière sur un réseau [32][50].

Dans la virtualisation complète, l'hyperviseur gère l'ensemble des requêtes des machines virtuelles [32] ce qui permet aux machines virtuelles de fonctionner sans aucune modification de leur noyau. Autrement dit, les machines virtuelles ne savent pas qu'elles s'exécutent de manière virtuelle.

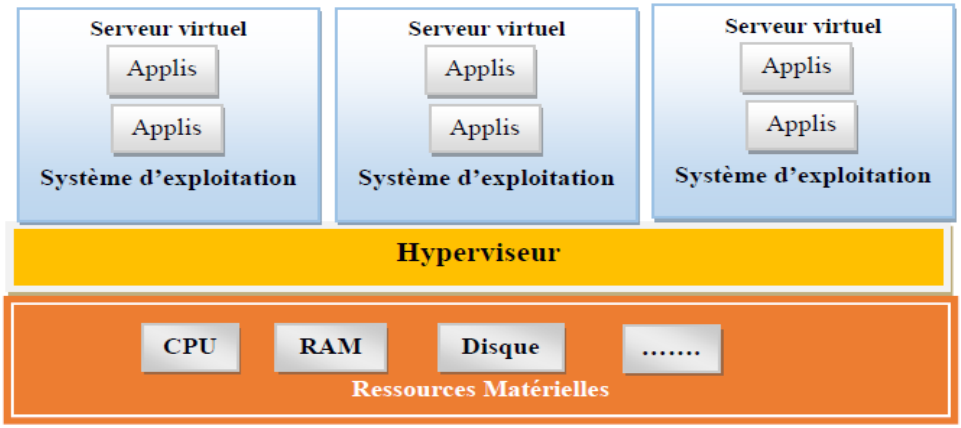

Figure 41 : Virtualisation complète

4.3.3 Para virtualisation

Le para virtualisation et la virtualisation complète sont assez proches. Elles s'appuient sur une couche hyperviseur, qui gère totalement l'interface avec les ressources matérielles, et sur laquelle on peut installer différents systèmes d'exploitation [31][49].

Le para virtualisation présente au système d'exploitation une machine générique spéciale (Figure 41), qui requiert donc des interfaces spéciales intégrées aux systèmes invités, sous la forme de drivers [49][50].

Le para virtualisation est une technique de virtualisation de plus bas niveau que l'isolation. Elle partage avec cette dernière la nécessité d'utiliser un OS modifié [49][50]. Plus précisément, en para virtualisation ce n'est plus seulement l'OS hôte qui doit être modifié mais également les OS appelés à s'exécuter sur les environnements virtuels [49].

Le cœur du para virtualisation est un hyperviseur fonctionnant au plus près du matériel, et fournissant une interface qui permet à plusieurs systèmes hôtes d'accéder de manière concurrente aux ressources [49]50]. Chaque système virtuel doit être modifié de façon à utiliser cette interface pour accéder au matériel [49].

Contrairement à l'isolation, plusieurs OS de familles différentes peuvent fonctionner sur un même serveur physique. Il est ainsi possible de faire fonctionner Linux, NetWare, Solaris (et d'autres) simultanément sur une même machine [49].

Chaque OS aura alors accès à ses propres périphériques de stockage, sa propre mémoire, sa ou ses propres interfaces réseau, son ou ses propres processeurs, chaque ressource matérielle virtualisée étant partagée avec les autres environnements [49]. La nécessité de petites modifications au système d'exploitation invité exclut le support de systèmes fermés, et en particulier de Microsoft Windows [31][49].

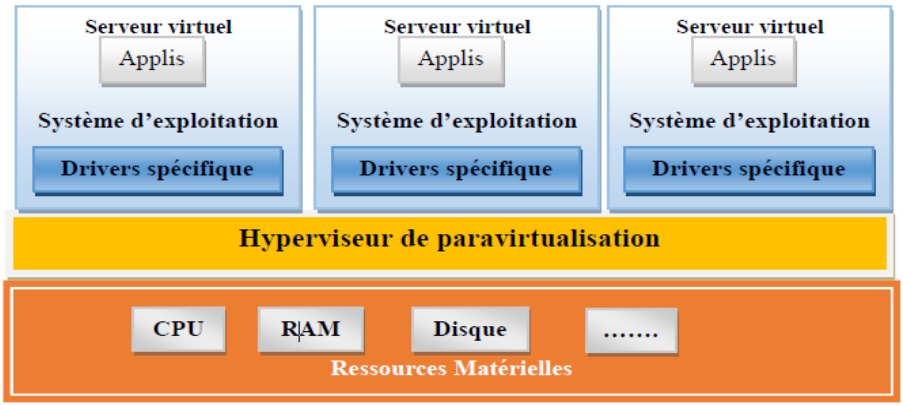

Figure 42 : Architecture de la paravirtualisation

4.4 Hyperviseurs

Un hyperviseur ou moniteur de machine virtuelle (VMM) est une «couche logicielle qui virtualise toutes les ressources d'une machine physique, définissant ainsi et supportant l'exécution de plusieurs machines virtuelles (VM)» [31][51]. L'efficacité des propriétés, le contrôle des ressources et l'équivalence d'un VMM ont été définis en 1974 [31]. Il existe deux types de VMM de type I et II [31][52]. Le premier type d'hyperviseur s'exécute directement au-dessus du matériel dans l'anneau de privilège le plus élevé (Figure 43) et contrôle toutes les machines virtuelles [31]. Ces hyperviseurs sont également appelés natifs et prennent en charge les machines virtuelles système classiques [31]. Quelques exemples d'hyperviseurs de type I sont VMware ESX, Hyper-V et Xen [31]. Tandis que le deuxième type d'hyperviseur (Figure 44), également appelé hébergé, s'exécute dans un système d'exploitation avec le système d'exploitation Ring of the Host et prennent en charge les machines virtuelles hébergées [31][52]. Quelques exemples d'hyperviseurs de type II sont VMware Workstation et VirtualBox.

Les hyperviseurs de type I ont des composants de gestion des ressources tels que la mémoire, le processeur et les E/S [31][52]. Par exemple, il existe un planificateur VM qui est utilisé pour la

gestion des ressources du processeur. Au contraire, les hyperviseurs de type II s'appuient sur le planificateur de processus de système d'exploitation, c'est-à-dire que chaque machine virtuelle en cours d'exécution est un autre processus de système d'exploitation [51][52].

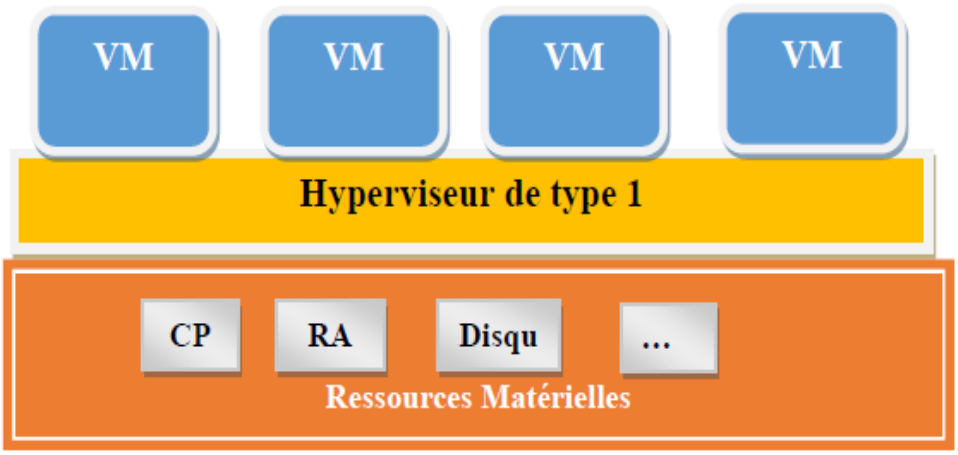

Figure 43 : Hyperviseur type I

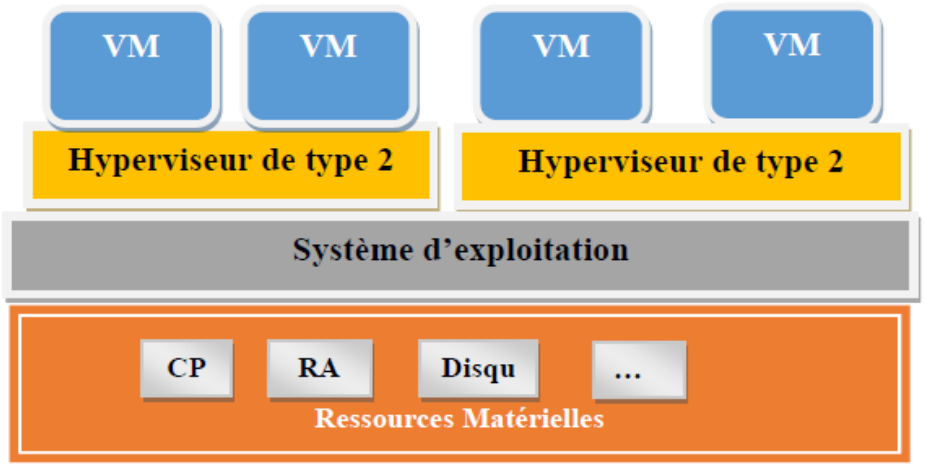

Figure 44 : Hyperviseur type II

4.5 Conclusion

La virtualisation est un domaine en pleine croissance, qui évolue très rapidement. Les entreprises peuvent s'en servir pour différents usages, aux besoins de leur fin. Les différentes solutions de virtualisation existantes utilisent des technologies variées, en fonction des buts du projet. Certaines technologies permettent de faire cohabiter plusieurs systèmes d'exploitation, d'autres cloisonnent un unique système en plusieurs compartiments indépendants. Certaines s'appuient sur les capacités du matériel pour améliorer les performances alors que d'autres nécessitent un système d'exploitation modifié pour cohabiter avec la solution de virtualisation. Ces technologies ont toutes leurs avantages et inconvénients, et il est important de faire le bon choix en fonction de l'utilisation que l'entreprise en fera.

En guise de conclusion, nous pouvons affirmer que la mise en place des serveurs virtuels est possible au sein de n'importe quelle architecture.

Chapitre 5 : Contribution à la sécurité d'une plateforme virtuelle

5.1 Introduction

La virtualisation des ressources informatiques, des centres de données aux systèmes stratégiques pour l'entreprise, en passant par les applications, a conduit à la réalisation des économies significatives en partageant l'espace de stockage, la capacité de l'unité centrale et l'espace mémoire. Malheureusement, la virtualisation est elle aussi sous le joug des menaces. Les logiciels malveillants sophistiqués et les attaques ciblées visent désormais les espaces virtualisés au-delà des frontières.

Cependant, les preuves démontrent que, malgré le fait que la virtualisation soit considéré comme une avenue commerciale majeure pour les années à venir, la migration vers le paradigme de la virtualisation est entravée par des préoccupations de la sécurité. Par exemple, les instituts financiers sont attirés par le cloud computing mais pour des raisons de sécurité, ils n'en sont encore qu'aux premiers stades d'adoption. Les récentes attaques contre le cloud, comme celle de 2014 où 50 millions de comptes utilisateurs de Dropbox ont été piratés [53], prouvent que la sécurité des données dans le cloud est devenue un sujet brûlant.

Pour que la virtualisation soit considérée comme une alternative viable, il doit offrir le même niveau de sécurité que les systèmes informatiques traditionnels. Pour atteindre cet objectif, une meilleure connaissance des mesures et des outils actuellement disponible pour combattre les actions malveillantes. Dans la littérature, il existe de nombreux travaux proposant des enquêtes sur les mécanismes de défense mis en place dans les infrastructures cloud [53].

Face à ce problème de sécurité, on va proposer une solution de contrôle d'accès aux infrastructures virtuelles open source, au début on va décrire les principales attaques de la plateforme. Ensuite nous proposons un firewall logiciel open source implémenté au niveau de l'infrastructure virtuelle

5.2 Menaces dans les systèmes informatiques virtuels

Les hyperviseurs s'exécutent sur les serveurs pour fournir la virtualisation des ressources physiques. La virtualisation des serveurs permettant d'exécuter plusieurs systèmes d'exploitation sur un même serveur physique sous la forme de machines virtuelles, dont chacune peut accéder aux ressources de calcul du serveur sous-jacent. La virtualisation des postes de travail offre la possibilité de réagir plus rapidement à l'évolution des besoins et des opportunités. Améliorer aussi les niveaux de service en distribuant de façon rapide et simple des postes de travail et applications virtualisées aux employés externalisés et sous-traitants, ou encore au personnel mobile travaillant sur iPad ou sur tablette Android. La virtualisation de réseau est la reproduction logicielle complète d'un réseau physique. Les applications du réseau virtuel s'exécutent de la même manière que si elles se trouvaient sur un réseau physique. La virtualisation du stockage isole les disques et les lecteurs Flash installés sur vos serveurs et les combine dans de vastes pools de stockage haut performances, qu'elle met à disposition sous forme de logiciel. La diversité de ces fonctionnalités rend la plates-forme de virtualisation plus vulnérables aux attaques que toute autre plate-forme informatique. Sa vulnérabilité peut être exposée par l'un de ses principaux composants : réseau, machines virtuelles, stockage et applications, qui servent de base à la catégorisation des attaques et à leurs implications.

5.2.1 Attaques réseau

Les machines virtuelle existant dans une plate-forme virtuelle sont connectées via un réseau qui fournit également des connexions avec les machines en dehors de la plate-forme. Un intrus peut attaquer un système de cloud à travers son réseau, ce qui peut à son tour détériorer la qualité des

services de cloud computing et peut même compromettre la confidentialité / confidentialité des données. On peut classer les attaques réseau types comme ci-dessous :

- **Analyse de port** : Un port sur un serveur peut être sondé pour vérifier l'état d'un service s'exécutant sur la machine cible. L'analyse de port nécessite l'accès au réseau hébergeant la machine cible. Il est utilisé pour exposer les vulnérabilités de la machine cible entraînant le déni de service. Les systèmes de détection d'intrusion ou pare-feu peuvent être utilisés pour détecter et entraver de telles attaques.
- **Botnet** : Est un réseau de bots informatiques (Figure 45), des programmes connectés à Internet qui communiquent avec d'autres programmes similaires pour l'exécution de certaines tâches [56][57].

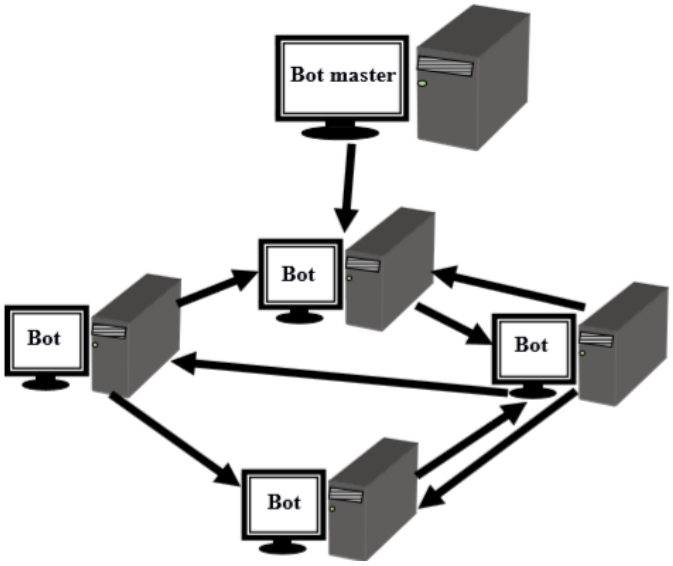

Figure 45 : Réseau de bots informatiques

- **Attaque par Spoofing** : Les attaques par usurpation d'identité dans un réseau usurpent l'identité d'entités à des fins malveillantes. Une attaque d'usurpation d'adresse IP peut remplacer l'adresse IP d'un paquet réseau par une adresse IP source falsifiée. De même, une attaque de spoofing DNS peut provoquer un serveur DNS pour renvoyer une adresse IP incorrecte en redirigeant le trafic réseau vers le système d'un attaquant. Un réseau virtuel peut être un victime de l'usurpation d'identité ARP, ce qui a provoqué l'intrusion d'une machine virtuelle de l'attaquant dans des paquets d'autres machines virtuelles [55]. L'antivirus basé sur le cloud ou les systèmes de détection d'intrusion peuvent être utilisés pour faire face à ces attaques.

5.2.2 Attaques basées sur la VM

Sur un système virtuel, les attaques par VM exploitent les vulnérabilités dans les machines virtuelles pour violer la protection des données et affecter les services de la plate-forme virtuelle. Plusieurs machines virtuelles hébergées sur un système entraînent plusieurs risques de sécurité [55]. De plus, diverses étapes de la gestion des machines virtuelles peuvent être utilisées pour lancer un grand nombre d'attaques de la plate-forme. D'après l'analyse de la littérature, nous pouvons décrire quelques types d'attaques VM :

- **Attaques par canaux auxiliaires :** Les attaques par canaux auxiliaires basées sur une VM peuvent extraire des informations concernant l'utilisation des ressources, les clés cryptographiques et d'autres informations d'une VM cible résidant sur la même machine physique que celle de l'attaquant [55]. Ces attaques peuvent exploiter les informations de

synchronisation à partir de ressources telles que le cache et la mémoire partagée [58]. Les contre-mesures pour les attaques par canal auxiliaire utilisent des mécanismes d'authentification, des algorithmes cryptographiques ou une exécution déterministe pour atténuer le risque de canaux auxiliaires.

- **Attaques créé par VM** : Un code malveillant peut être placé dans une image d'une machine virtuelle qui est ensuite répliquée lors de la création de machines virtuelles [58]. À cet égard, le système de gestion d'images des machines virtuelles doit fournir des filtres et des scanners pour détecter et résoudre les violations de la sécurité.
- **Attaques via la migration et la restauration d'une VM :** lorsqu'une VM active est migrée d'une machine physique hôte vers une autre machine physique (Figure 46), le contenu des fichiers VM devient vulnérable à diverses attaques [55]. Par exemple, le journal de l'état d'exécution conservé pour l'implémentation d'une restauration peut devenir accessible pendant la migration. Une configuration efficace des stratégies de sécurité ou des activités de suspension/reprise appropriées peut rendre la migration de la machine virtuelle plus sécurisée.

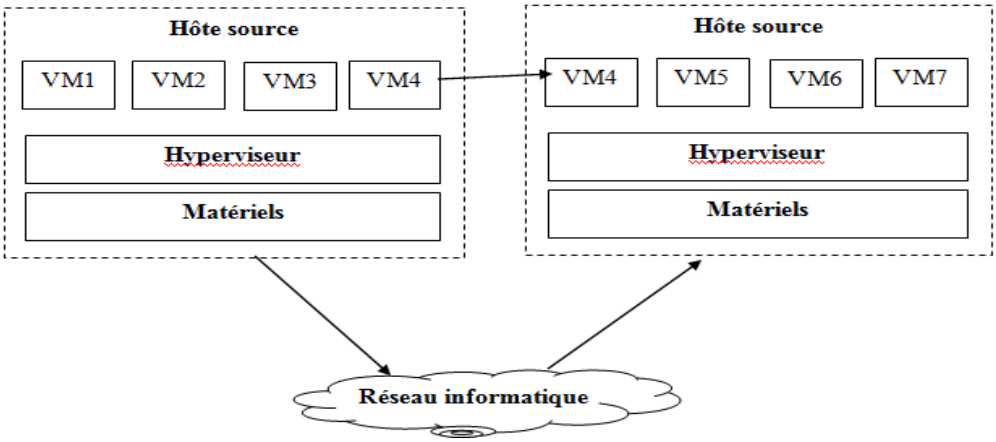

Figure 46 : Migration VM

5.2.3 Attaques basées sur l'espace de stockage

Un attaquant de l'extérieur ou même un initié malveillant peut voler des données privées stockées sur un support de stockage. Avec l'accès à des informations sensibles, un grand nombre de vulnérabilités peuvent être exploitées en manipulant des données si un mécanisme de surveillance strict n'est pas mis en œuvre [55] [59].

5.2.4 Attaques basées sur applications

Les applications s'exécutant sur un nuage peuvent être exposées à diverses attaques en injectant du code qui peut tracer des chemins d'exécution et exploiter cette information à des fins malveillantes. De même, les protocoles implémentés pour fournir des services sur un système cloud sont vulnérables aux attaques et toutes les applications en cours d'exécution peuvent les utiliser comme source d'intrusion. De plus, sur un système en nuage, les composants architecturaux partagés peuvent être exploités par une application comme source d'exécution d'activités malveillantes [55].

5.3 Protection et surveillance d'une plateforme virtuelle

Une infrastructure informatique physique ou virtuelle nécessite la protection et la mise en place des mesures de sécurité. Elle regroupe le système d'information qui est défini par l'ensemble des

données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Il représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger.
Dans cette section, nous travaillons sur la protection d'accès à l'infrastructure virtuelle et la surveillance de flux de trafic entre et sort de celle-ci, par la mise en place d'un firewall et d'un système de détection d'intrusion open source.

5.3.1 Firewall logiciel open source (Netfilter/iptable)

Les Firewalls font partie intégrante de la sécurité d'un système ou d'un réseau informatique. Ils permettent de limiter l'accès à des applications ou des parties de réseau grâce à des règles (statiques ou dynamiques) définies par la politique de sécurité mise en place par les administrateurs réseaux ou responsables de la sécurité du système d'information d'une entreprise.
Iptables est un utilitaire inclus dans les distributions linux qui permet de piloter Netfilter. Ce dernier est implémenté dans le noyau linux [59]. Cela inclut le filtrage de paquets ou effectuer du NAT [59]. En clair, iptables permet de créer des règles de pare-feu pour filtrer les paquets entrant/sortant en direction des services réseaux et processus d'une machine (Figure 47).

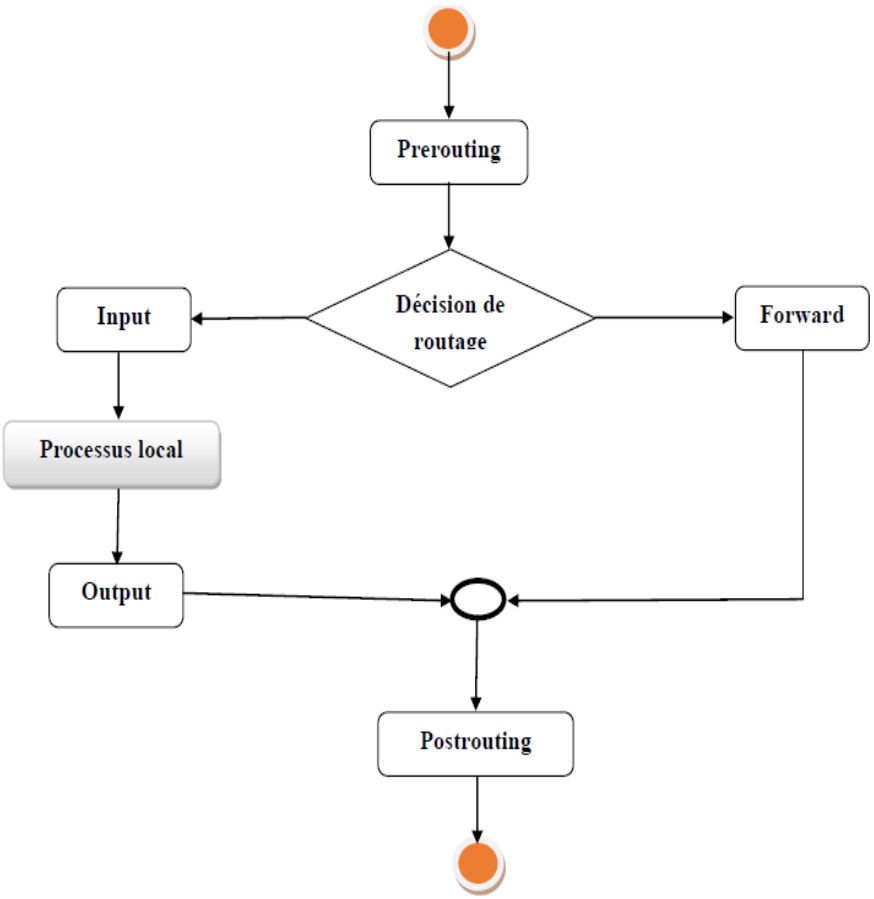

Figure 47 : Points d'accrochage de l'Iptable/Netfilter

5.3.2 Système de détection et prévention SURICATA

Le système de détection d'intrusion (IDS) analyse l'en-tête et la partie payload du paquet pour le comparer à toutes les anomalies trouvées par rapport au trafic normal. Ceci est en contraste avec un pare-feu qui filtre le trafic réseau en examinant les en-têtes de paquets circulant dans les ports réseau. Pour le trafic anormal, un IDS tente d'identifier le modèle contre les menaces courantes et alerte l'administrateur réseau. Un système de prévention des intrusions (IPS) fonctionne exactement comme un IDS, mais il peut également rejeter les paquets ou mettre fin à la connexion.

Le système de prévention des intrusions Suricate est un moteur basé sur des règles qui prend en charge la détection et la prévention des intrusions en surveillant le trafic réseau [60]. Il génère des alertes pour l'administrateur système si une activité suspecte est effectuée sur le réseau (Figure 48). Sur les architectures hautes performance, il peut évoluer efficacement en exploitant la fonctionnalité de multithreading sur plusieurs processeurs ou cœurs. Son système de détection peut identifier les protocoles, ce qui permet aux utilisateurs d'obtenir facilement une sécurité basée sur des protocoles au lieu de ports.

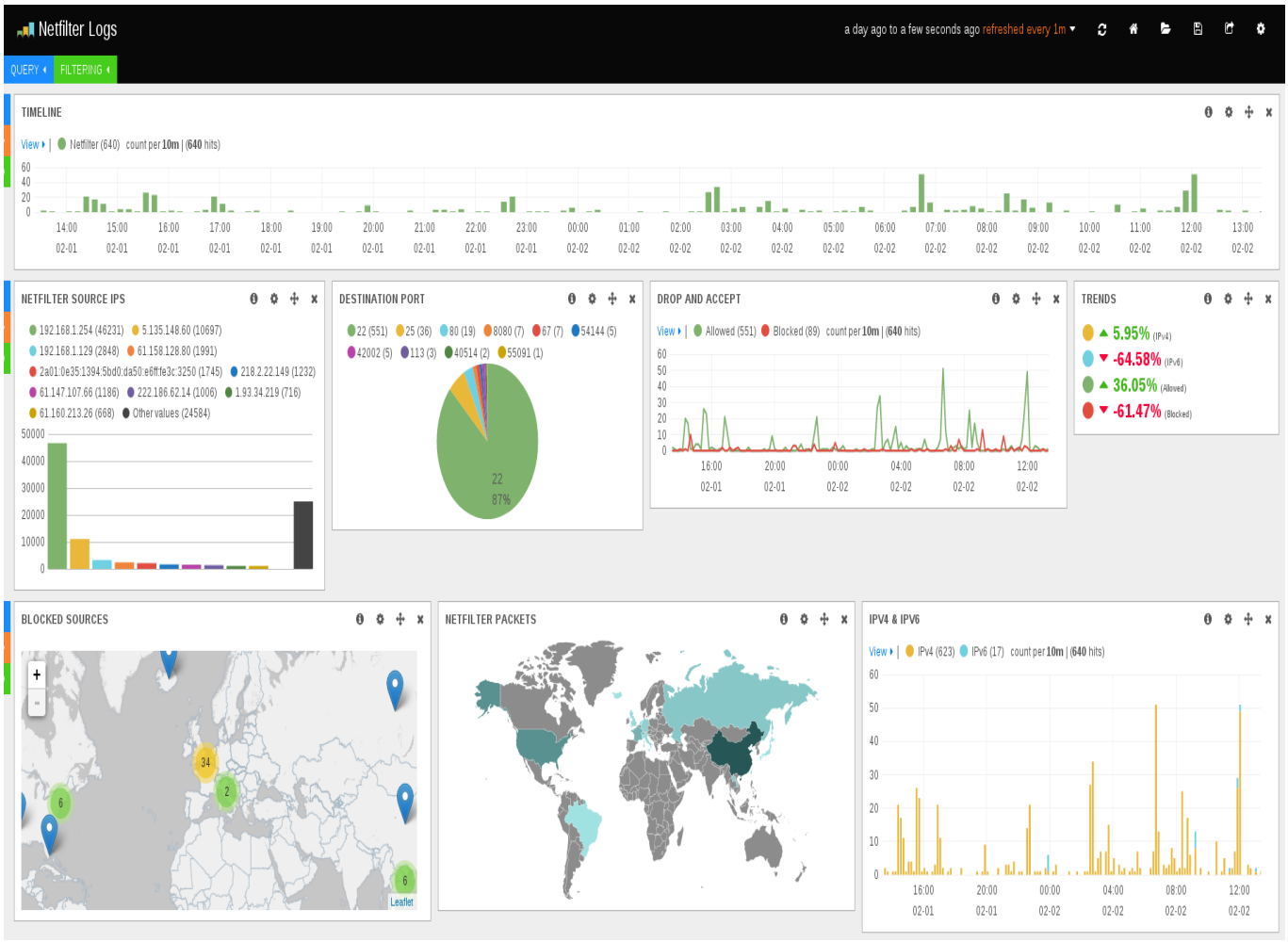

Figure 48 : Interface des alertes Netfilter Logs [61]

5.3.3 Architecture de l'approche

Pour pouvoir tester expérimentalement notre approche, nous proposons une plateforme de virtualisation open source reposant sur la solution Kernel-based Virtual Machine (KVM). Le choix s'est porté sur cette technologie car, Il permet une virtualisation matérielle et donc une accélération de la virtualisation de système d'exploitation. C'est un système optimisé pour la virtualisation de serveur. Il est également plus performant en consommation de processeur et est très intéressant grâce à sa meilleure compatibilité avec des systèmes d'exploitation plus anciens et facile à mettre en place. Dans cette approche nous contrôlons le flux de trafic entrant/sortant de l'infrastructure par un firewall implémenté sur une machine virtuelle utilisée comme une Front Gateway pour l'infrastructure (Figure 49).

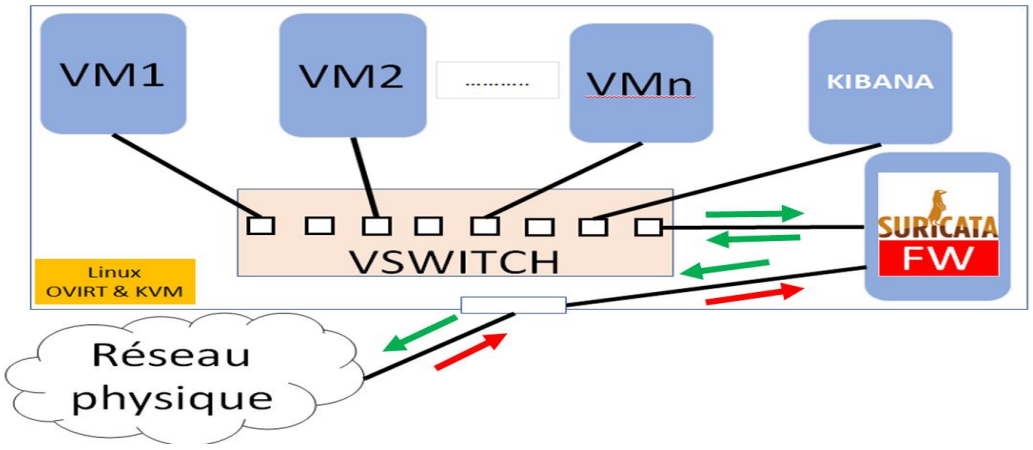

Figure 49 : Architecture de la protection d'accès à la plateforme virtuelle

La Front Gateway représente le point d'entrée de notre infrastructure, elle garantit la gestion et le traitement nécessaire pour distribuer le trafic légitime sur nos VMs. Sur cette machine on installe IDS/IPS Suricata et on active le firewall Netfilter/Iptables. Par la suite, nous déployons sur chaque virtuelle machine un firewall. Enfin, nous mettons une machine qui permet de surveiller, analyser et visualiser les activités (logs) du front Gateway en temps réel sur une interface graphique simple pour adapter les règles de ce firewall, pour cela on utilise l'analyseur des logs la pile ELK «Elasticsearch + Logstash + Kibana »

5.3.4 Surveillance de la plateforme virtuelle avec ELK

La gestion des logs système constituent une partie essentielle de la gestion de la sécurité réseau et de l'administration du système d'information de l'entreprise. Les logs contiennent des informations importantes, notamment l'état actuel et l'historique du système qui font référence à différents évènements de sécurité, qui se produisent dans le système. Les journaux (logs) sont utilisés à différentes fins, telles que l'enregistrement des activités de l'utilisateur, le suivi des tentatives d'authentification et d'autres événements de sécurité. En raison de l'augmentation du nombre de menaces contre les réseaux et les systèmes, le nombre de journaux de sécurité augmente. Cependant, l'analyse et la surveillance des journaux système deviennent une tâche essentielle pour maintenir leur disponibilité et leur sécurité. L'analyse du log peut aider les experts en sécurité à prendre conscience de la situation actuelle des serveurs et à réagir à des événements imprévus à temps, tels qu'un dysfonctionnement ou une intrusion.

Vues le volume des logs et les alertes générés par l'IPS SURICATA, nous proposons une interface graphique composé d'Elasticsearch, Logstash et Kibana qui permet d'afficher et classer ces messages et d'analyser logs afin de réécrire et redéfinir les règles de sécurité au niveau de l'IPS (Figure 50).

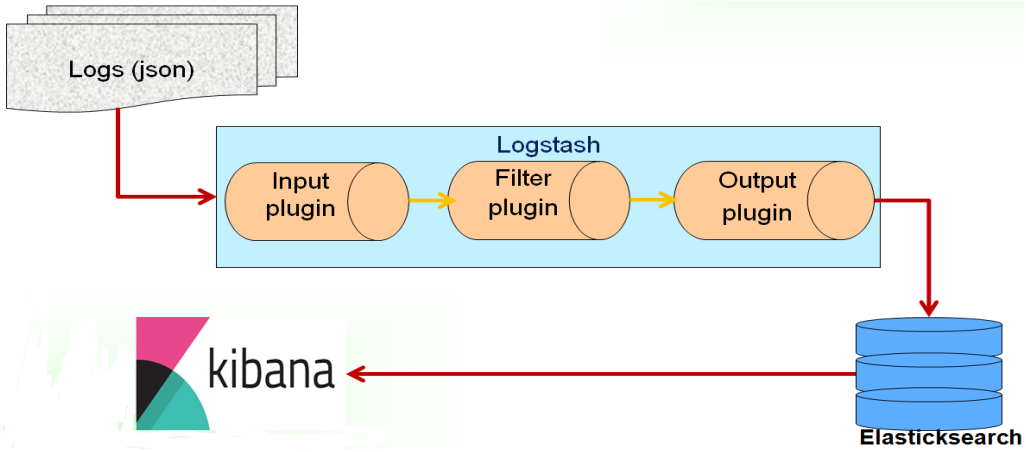

Figure 50 : Processus de surveillance par le stack ELK

- **Logstash :** Logstash est un pipeline de données qui permet de collecter, d'analyser une grande variété de données et événements structurés et non structurés générés par différents systèmes. Il fournit des plugins pour se connecter à différents types de sources d'entrée et de plates-formes,
  est conçu pour traiter efficacement les logs, les événements et les sources de données non structurées pour distribution dans une variété de sorties avec l'utilisation de ses plugins de sortie, à savoir le fichier, stdout (en tant que sortie sur une console exécutant Logstash), ou Elasticsearch. Dans notre approche, on utilise Logstash pour analyser et extraire les évènements généré par L'IPS Suricata dans une infrastructure virtuelle puis les transferts vers la base d'indexation Elastisearch sous la forme JSON (Figure 51).

```
input {
        file {
          path => ["/var/log/suricata/eve.json"]
          sincedb_path => ["/var/lib/logstash/"]
          codec =>   json
          type => "SuricataIDPS"
   } }

filter {
        if [type] == "SuricataIDPS" {
                date {
                match => [ "timestamp", "ISO8601" ]
          }
        ruby {
                code  => "if event['event_type'] == 'fileinfo';
                event['fileinfo']['type']=event['fileinfo']['magic'].to_s.split(',')[0]; end;"
         }       }
      if [src_ip]  {
        geoip {
                source => "src_ip"
                target => "geoip"
```

```
                database => "/opt/logstash/vendor/geoip/GeoLiteCity.dat"
                add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
                add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
          }
        mutate {
                convert => [ "[geoip][coordinates]", "float" ]
        }
        if ![geoip.ip] {
                if [dest_ip]  {
                        geoip {
                        source => "dest_ip"
                        target => "geoip"
                        #database => "/opt/logstash/vendor/geoip/GeoLiteCity.dat"
                        add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
                }
        mutate {
                convert => [ "[geoip][coordinates]", "float" ]
         } }  } }}
output {
          elasticsearch {
              host => @IP   #la VM sur laquelle installé ElasticSearch et KIBANA
 } }
```

Figure 51 : Fichier de configuration Logstash

- **Elasticsearch :** Elasticsearch stocke les données de log transférées sous la forme de documents JSON (JavaScript Object Notation) structurés et fournit une plate-forme de recherche et d'analyse. Les données de journal d'un module spécifique peuvent être distinguées par le nom de conteneur unique inclus dans l'en-tête du journal et envoyé les résultats à l'interface graphique Kibana qui constitue une table de bord pour les gens de sécurité afin de prévenir dans le temps.

- **Kibana** : Est une plate-forme d'analyse et de visualisation open source conçue pour fonctionner avec Elasticsearch. Nous utilisons Kibana pour rechercher, afficher et interagir avec les logs stockées dans les index Elasticsearch. Nous pouvons facilement effectuer une analyse avancée des logs et visualiser le contenu de ces logs dans divers formats : tableaux, représentation graphiques et carte de géolocalisations. Kibana facilite la compréhension de gros volumes des journaux. Son interface simple basée sur un navigateur nous permet de créer et de partager rapidement des tableaux de bord dynamiques qui affichent les modifications apportées aux requêtes Elasticsearch en temps réel.

- **Tableau de bord ELK :** Apres l'installation de la pile ELK, nous pouvons personnaliser le tableau de bord en fonction de nos besoins en termes de sécurité, par exemple nous pouvons créer un compteur qui compte les nombres des alertes générer par l'IDS/IPS Suricata. Par la suite on peut filtrer toutes les tentatives de connexions et indiquant les noms d'utilisateurs, adresses IP et le pays de l'attaquant. Pour donner une idée claire à l'administrateur sur les

attaquants afin de prendre les mesures de sécurité adéquats. En fin la connaissance des attaquants nous aidons d'améliorer le mécanisme de sécurité et connaitre les capacités dont ils disposent pour affiner nos méthodologie afin de traiter les différents types d'attaques utilisées contre notre systèmes.

5.4 Conclusion

Dans ce chapitre nous nous intéressons aux problèmes de contrôle d'accès aux espaces virtuels, et compte tenu de la multitude des attaques interne ou externe de la plateforme virtuelle, nous avons vu qu'il était préférable de disposer d'un front firewall qui permet de contrôle de filtrer le flux entrant/sortant, et au même temps, nous mettons un IDS/IPS sur la même machine afin d'analyser pour détecter, alerter et analyser ces intrusions. Cependant, si l'on a mis en place une sonde, cela ne permet pas pour autant de sécuriser l'infrastructure, mais aide à mieux en percevoir ses défauts, quitte à améliorer par la suite les règles de sécurité des pare-feu et à utiliser davantage d'outils et technologies de sécurité. De plus un tel outil nous permet d'obtenir des statistiques sur les tentatives d'intrusion, réussies ou non, et de les tracer afin d'obtenir des pistes de prévention, voire de répression.

Chapitre 6 : Évaluation comparative des performances du système de détection d'intrusion : Suricata et Snort

6.1 Introduction

La sécurité du système d'information de l'entreprise est un gros problème aujourd'hui. Malheureusement, les hackers et les intrus ont fait de nombreuses tentatives réussies, sur des réseaux des entreprises de haut niveau [62]. Malgré des nombreuses méthodes ont été développées pour sécuriser l'infrastructure de réseau et de communication sur Internet, dont l'utilisation de pare-feu, le cryptage et les réseaux privés virtuels. Les attaquants agissent comme des utilisateurs normaux, génèrent des données et cachent leurs activités malveillantes. Ils savent que de nombreux mécanismes de sécurité ne peuvent pas protéger les systèmes informatiques en raison de la grande quantité de données stockées, des problèmes d'évolutivité ou du manque de capacités de détection. Les entreprises doivent surveiller leurs systèmes informatiques pour détecter les activités malveillantes et effectuer des analyses pour différencier les activités des utilisateurs malveillants et les activités légitimes afin de protéger leurs infrastructures. La détection des activités malveillantes nécessite des systèmes de détection d'intrusion (IDS) pour auditer leurs systèmes et détecter d'éventuelles intrusions [65].

Les systèmes de détection d'intrusion (IDS) sont en train de devenir l'un des composants essentiels de tout réseau d'entreprise. Les IDS sont conçus pour détecter toute intrusion ou tout trafic hostile sur un système informatique. Face au besoin urgent de tels systèmes de détection, les entreprises investissent dans la mise au point d'un système IDS plus efficace. Les systèmes de détection d'intrusion peuvent être mis en œuvre sous forme de matériel ou de logiciel [63]. L'IDS logiciel est plus configurable et facile à mettre à jour alors que le matériel est conçu pour gérer une grande quantité de trafic mais plus coûteux et nécessite plus de maintenance. Il est donc nécessaire d'étudier la performance de l'IDS logiciel disponible. En général, les systèmes de détection d'instructions se divisent en deux catégories principales : Systèmes basés sur le réseau et systèmes basés sur l'hôte [64][65].

Snort et Suricata sont deux systèmes de détection d'intrusion (IDS) qui analysent le trafic réseau, analysent les protocoles de niveau supérieur et signalent la présence d'activités réseau malveillantes ou indésirables [66].

Dans ce chapitre, nous nous sommes concentrés sur les IDS basés sur les signatures, l'accent étant mis sur l'évaluation des performances dans les réseaux à grande vitesse. Notre objectif est de fournir une comparaison détaillée entre les deux IDS avec un trafic à grande vitesse.

6.2 Système de détection d'intrusion : Suricata et Snort

6.2.1 IDS Snort

Snort est un système de détection d'intrusion open source, capable de consigner des paquets, d'analyser le trafic et de détecter les intrusions à l'aide de signatures. En plus de l'analyse de protocole, Snort effectue diverses correspondances de contenu sur les paquets réseau en recherchant des modèles d'attaques et de sondes connues [62]. Snort utilise un langage de règles flexible, permet aux utilisateurs de décrire le trafic qui doit être collecté ou transféré et dispose d'un moteur de détection utilisant une architecture de plug-in (Figure 52) modulaire [62][67]. Le système d'alerte en temps réel fourni par Snort incorpore des mécanismes d'alerte pour syslog, les fichiers spécifiés par l'utilisateur, les sockets UNIX ou les messages WinPopup sur les clients Windows à l'aide du protocole SMB[62]. Snort fonctionne sur diverses plates-formes: Linux (i386, Sparc, M68k / PPC,

Alpha) OpenBSD (i386, Sparc, M68k / PPC), FreeBSD (i386), NetBSD (i386, M68k / PPC),Solaris (i386, Sparc), SunOS 4.1.X (Sparc), MacOS X Server (PPC) et Win32 (i386)[62].

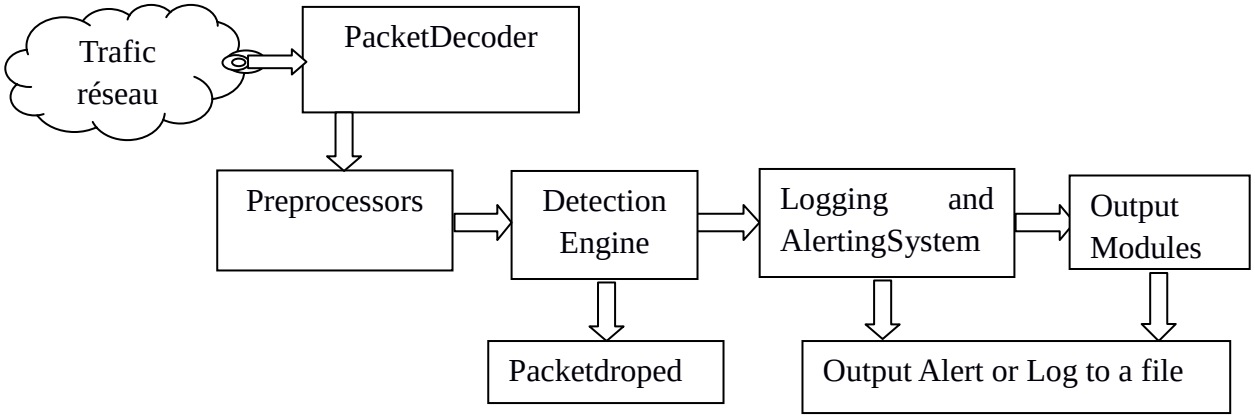

Figure 52 : Architecture de l'IDS Snort.

6.2.2 IDS Suricata

Suricata est un système IDS / IPS basé sur des règles source qui utilise des ensembles de règles développés en externe pour surveiller le trafic réseau et fournir des alertes aux administrateurs de la sécurité lorsqu'une éventuelle attaque est détectée. Il a été développé par la fondation Open Information Security (OISF). Suricata utilise un format de règle différent de celui de snort. Suricata se distingue également de Snort en prenant en charge plusieurs threads dans un environnement à plusieurs processeurs afin d'accélérer le traitement des paquets dans les réseaux à haut débit [68]. Le système hautes performances Suricata IDS pour la surveillance de réseau a été développé comme une amélioration open source pour le système Snort populaire disponible depuis plus d'une décennie. Suicata présente plusieurs fonctionnalités, notamment le multi-threading [68] (Figure 53) pour améliorer la vitesse de traitement. Suricata améliore également Snort dans l'analyse basée sur les états, ce qui est particulièrement important pour le trafic HTTP. Il est également basé sur des signatures mais intègre des techniques révolutionnaires. Ce moteur intègre un normalisateur et analyseur http qui fournit un traitement très avancé des flux HTTP, permettant de comprendre le trafic au septième cMMouche du modèle OSI[68].

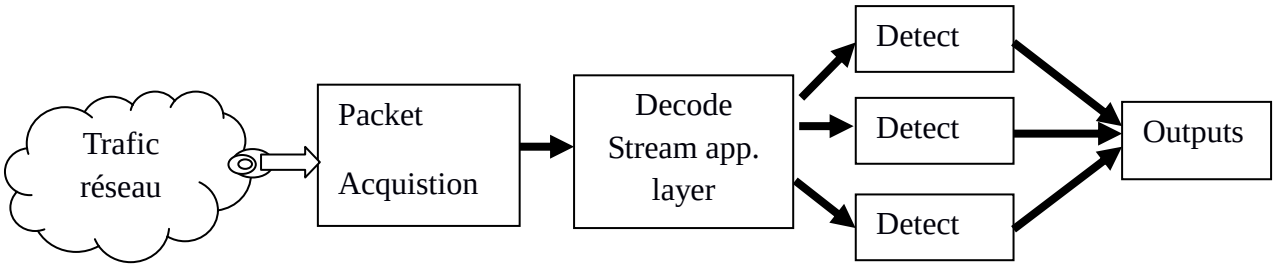

Figure 53 : Architecture de l'IDS Suricata

6.3 Travaux relatifs

Le volume de trafic réseau augment, le nombre et la complexité des attaques réseau augment avec une très grande vitesse, ce rend la détection d'intrusiondifficile à réaliser. Cependant, il devient de plus en plus difficile pour un système de détection d'intrusion basé sur la signature de faire face aux menaces actuelles [69]. Lorsqu'un système de détection d'intrusion échoue de générer d'alarme, le résultat pourrait être la compromission de données critiques au sein de l'infrastructure réseau. De

plus, si le système de détection d'intrusion génère trop de fausses alarmes [70], les opérateurs surveillant le système peuvent devenir insensibles aux alertes et ignorer une alerte réelle à l'avenir.

Les alertes générées par un système de détection d'intrusion peuvent se produire dans quatre situations vraies positives, vraies négatif, faux positif et faux négatif [69][71]. Le vrai positif est lorsque le moteur de détection génère une alerte basée sur l'identification correcte d'une menace potentielle [73]. Le vrai négatif est lorsqu'un moteur de détection ne génère pas d'alerte pour le trafic normal. Le faux positif se produit lorsqu'un moteur de détection génère une alerte pour un trafic non malveillant. La plus dangereuse des conditions est le faux négatif lorsqu'un moteur de détection n'alerte pas sur le trafic malveillant [71], lui permettant ainsi d'entrer sur le réseau sans préavis. La précision de la détection d'intrusion peut être mesurée par le nombre de faux positifs et de faux négatifs générés par le moteur de détection [72].Des expériences ont été réalisée sur les systèmes de détection Snort et Suricata pour déterminer la performance d'utilisation des ressources matériels (CPU,Mémoire) et la précision de détection des intrusions [74]. Les expériences ont été effectuées sur deux ordinateurs hôtes différents avec des spécifications de CPU, de mémoire et de carte réseau différentes. Leurs résultats ont montré que Suricata nécessitait une puissance de traitement supérieure pour fonctionner correctement par rapport à Snort. De plus, les résultats ont montré qu'avec une puissance de traitement supérieure, Suricata pouvait détecter avec précision le trafic malveillant sur le réseau et que son jeu de règles était efficace [75].

KittikhunThongkanchorn, SudsanguanNgamsuriyaroj, Vasaka Visoottiviseth [76] ont étudié les performances et la précision de la détection de trois systèmes de détection d'intrusion open source: Snort, Suricata et Bro. Ils ont évalué tous les systèmes utilisant différents types d'attaques, y compris les attaques par déni de service, les attaques DNS, les attaques FTP, les attaques par analyse du port et les attaques SNMP. Les expériences ont été exécutées sous différents débits de trafic et différents ensembles de règles actives. Les mesures de performance utilisées sont l'utilisation de la CPU, le nombre de paquets perdus et le nombre d'alertes. Les résultats ont montré que chaque type d'attaque avait des effets importants sur les performances de l'IDS. Mais Bro a montré de meilleures performances que les autres systèmes IDS lorsqu'il a été évalué sous différents types d'attaque et en utilisant un ensemble spécifique de règles. Les résultats ont également indiqué la perte de précision lorsque les trois outils IDS activent l'ensemble des règles.

6.4 Méthodologie de l'étude comparative des deux IDSs

Les scénarios expérimentaux ont été conçus pour faire des observations et prendre des mesures. Cette étude a démontré des comparaisons de performances quantitatives rigoureuses des deux IDS. L'expérience est un nombre d'essai qui comparait la précision de détection de Snort et de Suricata à une vitesse de réseau de 10 Gbps et avec sept types différents de trafic malveillant. Les sept types de trafic malveillant ont été choisis car les règles pouvaient être appliquées de manière cohérente à Snort et à Suricata. De plus, ils sont les types les plus courants et couvrent un bon nombre d'attaques. Les expériences ont comparé les performances des deux IDS en mesurant le pourcentage de CPU, l'utilisation de la mémoire et le taux de perte de paquets du réseau. La précision de détection des jeux de règles de Snort et Suricata est mesurée par le trafic malveillant dans un environnement d'expérience contrôlé et comparée au nombre d'alarmes fausses positives, fausses négatives et vraies positives déclenchées par chaque IDS. Le trafic réseau normal pour les expériences a été produit à l'aide de générateurs de trafic réseau open source (Ostinato, NMAP et NPING). Ces outils peuvent générer un trafic réseau allant jusqu'à 20 Gbps. Le trafic malveillant a été généré à l'aide du framework Metasploit sur Kali Linux.

Les règles par défaut de Snort et Suricata ont été utilisées pour les expériences. En outre, le trafic réseau légitime et le trafic malveillant ont été générés et un trafic combiné est fourni en entrée à Snort et Suricata. Certaines des questions que nous devions poser étaient les suivantes. Quel IDS offre des performances supérieures lors du traitement d'un trafic réseau allant jusqu'à 10 Gbps? L'architecture de l'IDS a-t-elle un impact sur cela? Quelles sont les différences de taux de perte de paquets entre Snort et Suricata lorsque l'utilisation du processeur, de la mémoire et du réseau augmente ?

Le réseau de test illustré à la figure 54 a été construit à l'aide KVM (Kernel-based Virtual Machine). Cinq machines virtuelles (VM) ont été créées. En fonction des besoins de chaque expérience, des paquets réseau légitimes et malveillants ont été générés à différentes vitesses avec des outils générateurs de trafic réseau. Les cinq machines virtuelles ont été connectées via un commutateur virtuel (OpenVswitch) utilisant des liaisons Ethernet à 10 Gbps. Le réseau expérimental était constitué de machines virtuelles hautes performances exécutant Snort et Suricata. Les dernières versions disponibles de Snort (v2.9.6.12) et de Suricata (v4.0.5) ont été utilisées pour les expériences.

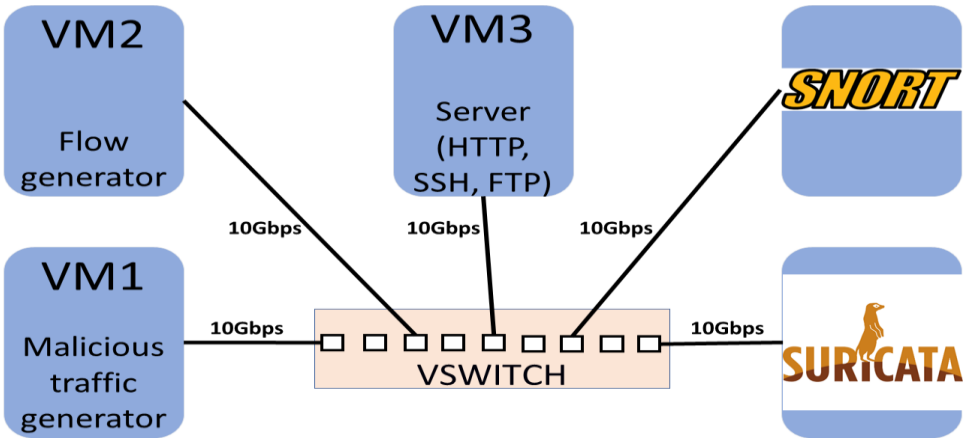

Figure 54 : Réseau de test : Linux OVIRT et KVM

6.5 Expérimentations et discussions

Les scénarios d'expérimentation ont été planifiés et configurés pour comparer les performances de Snort et de Suricata sur des VM identiques utilisant des règles identiques et dans les mêmes conditions de test.

6.5.1 Premier scénario : Mesure des performances de point de vue de CPU et de mémoire

Cette expérience a permis d'observer les performances en temps réel de Snort et Suricata lors du traitement à une vitesse de réseau légitime de 10 Gbps à partir d'un générateur de trafic réseau légitime (Ostinato). La première expérience visait à comparer les performances de Snort à celles de Suricata. Pour obtenir des résultats précis, le scénario de l'expérience a été testé avec une taille de paquets de 1 470 octets pour TCP, UDP et ICMP. Ces paquets ont été injectés aux deux IDS avec une vitesse de réseau de 10 Gbps. Chaque système IDS a été installé séparément sur des machines virtuelles identiques avec des paramètres de performance et un ensemble de règles par défaut. Un certain nombre d'outils ont été utilisés pour observer et enregistrer les mesures de CPU, mémoire, utilisation du réseau et taux de perte de paquets. Il s'agit de l'outil Collectl, Framework Metasploit, top, dstat, des journaux Snort, tcpdump, IPTRAF, nmap. Les paquets suivants ont été injectés en tant que trafic de fond compris entre 1 Gbps et 10 Gbps, comme suit. (1) 1 000 000 de paquets UDP avec un débit de 500 paquets/s, chaque taille de paquet étant de 1 470 octets (2) 1 000 000 de

paquets TCP avec un débit de 500 paquets/s, chacun la taille de paquet étant de 1 470 octets (3) 1 000 000 de paquets ICMP avec un débit de 1 000 paquets/s, chaque taille de paquet étant de 1 470 octets. Les données recueillies de l'expérience ont été enregistrées et observées. Il existe des options pour injecter les paquets normalement ou en rafales. L'option de configuration pour le trafic en rafale nécessite le nombre de rafales, le nombre de paquets par rafale et le nombre de rafales par seconde. Nous avons choisi le trafic normal en spécifiant le nombre de paquets par seconde et le nombre de paquets. En outre, plusieurs instances d'injection de paquets peuvent être effectuées. Les données recueillies lors de l'expérience ont montré que l'utilisation du processeur Suricata était supérieure à celle de Snort tout en traitant le même trafic réseau à 10 Gbps (figure 55).

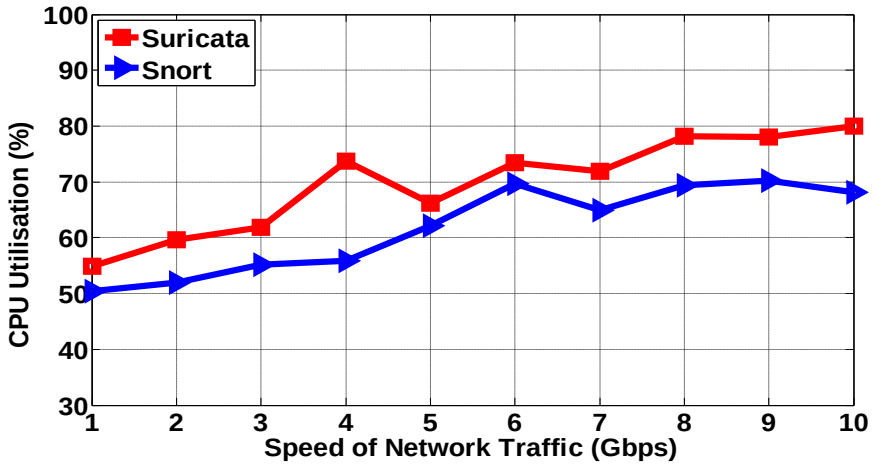

Figure 55 : Pourcentage d'utilisation de CPU par Snort et Suricata

Selon les résultats obtenus on trouve que l'utilisation du processeur par Snort est prouvé moindre que celui de Suricata. Les données de performance collectées ont montré que l'utilisation de la mémoire de Suricata était supérieure à celle de Snort (Figure 56). L'utilisation moyenne de la mémoire de Suricata a augmenté de 3,3 Go à 1 Gbps et a continué d'augmenter à un débit variable jusqu'à un maximum de 3,9 Go lors du traitement à Vitesse du réseau de 10 Gbps.

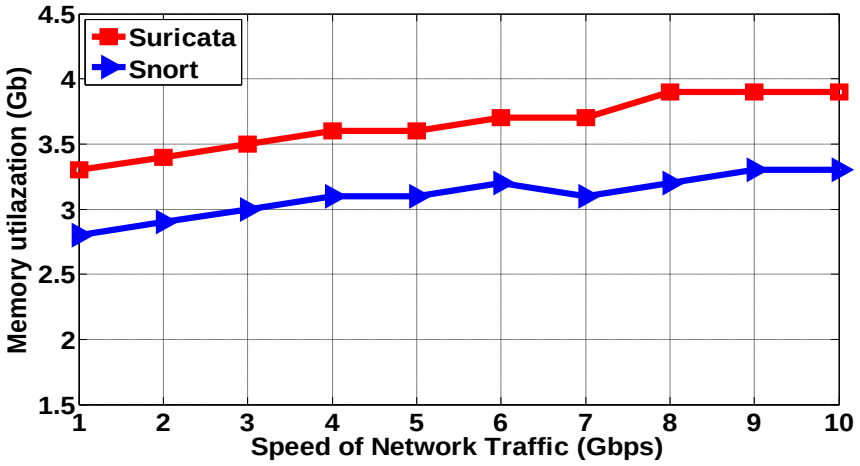

Figure 56 : Utilisation de la mémoire en Gbytes par Snort et Suricata

L'utilisation moyenne de la mémoire par Snort était comparativement moindre, à partir de 2,8 Go à 1 Gbps et continuait à fonctionner avec une utilisation réduite de la mémoire à toutes les vitesses du réseau, menant à une utilisation de la mémoire de 3,3 Go lors du traitement à une vitesse de réseau de 10 Gbps. L'utilisation de la mémoire de Suricata est davantage liée à l'architecture multithread.

Lorsque la vitesse du trafic réseau commence à augmenter, l'utilisation du processeur et de la mémoire commence également à augmenter pour les deux IDS. Les fichiers journaux Snort et Suricata montraient statistiquement le processus avec des paquets à des vitesses de réseau variables ainsi que le volume d'utilisation du processeur et de la mémoire. Le moteur de traitement des paquets de Snort était plus lent que le moteur de traitement de Suricata. La même quantité (1 000 000 de paquets) de paquets UDP, TCP et ICMP a été injectée dans les deux IDS pendant la même durée. Suricata montre d'excellentes performances par rapport à Snort lors du traitement de grandes quantités de paquets (Figure 57).

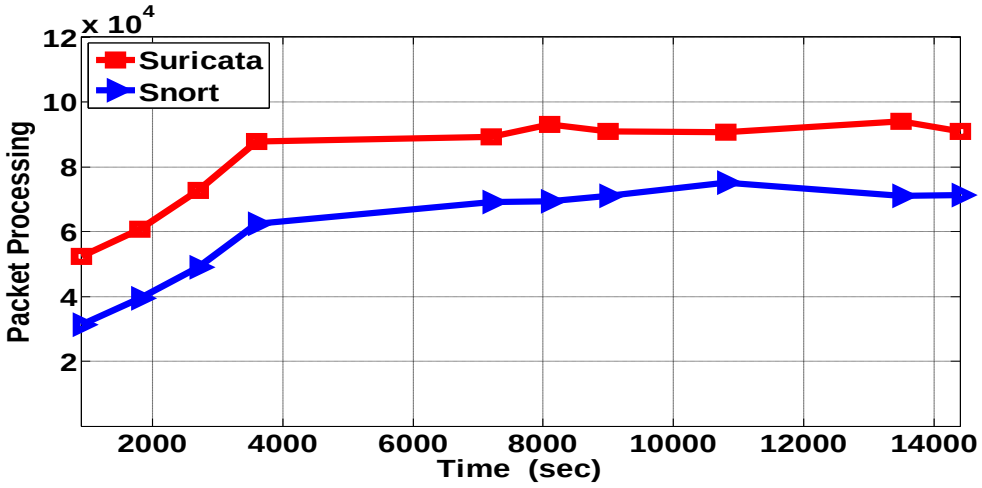

Figure 57 : Taux de traitement des paquets par Snort et Suricata par seconde

Comme le montre la figure 58, les performances du réseau sont restées un problème car Snort a abandonné plus de paquets à une vitesse de réseau de 10 Gbps par rapport à Suricata. L'expérience a été réalisée comme auparavant, et la valeur moyenne est prise parmi les trois valeurs pour UDP, TCP et ICMP pour les deux IDS.

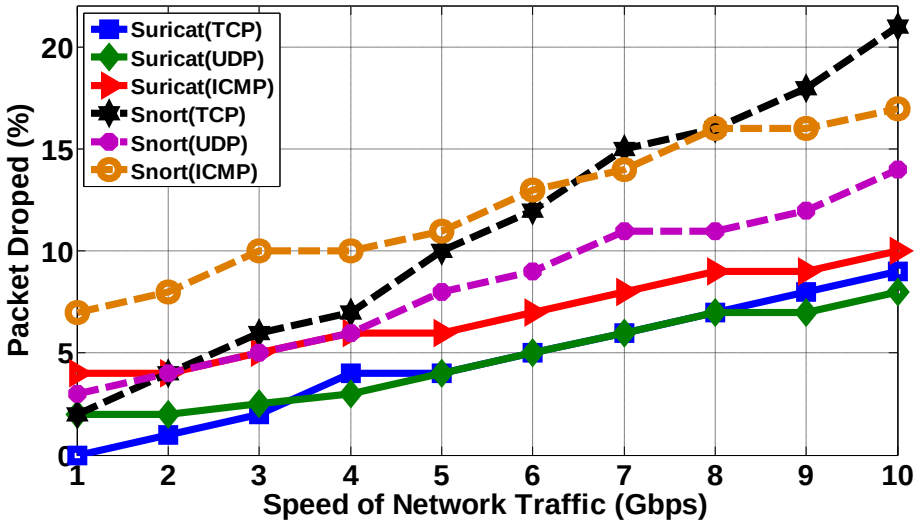

Figure 58 : Le nombre moyen de paquets chute à une vitesse de réseau variable

6.5.2 Deuxième scénario : Mesure de la précision des moteurs de détection

Cette expérience avait pour but de déterminer la précision avec laquelle le jeu de règles Snort et Suricata avait inspecté le trafic réseau à 10 Gbit/s afin de classer correctement le trafic légitime et le

trafic malveillant. Les mesures énumérées ci-dessous ont été utilisées pour mesurer la précision de détection des deux IDS.

- Taux de faux positifs (FPR)

C'est la probabilité que l'IDS déclenche une alarme en l'absence d'intrusion. Cela a été mesuré en pourcentage de FPR. Cela déclenche un taux excessif de fausses alarmes positives lorsqu'elles ne font pas la différence entre le trafic réseau légitime et le trafic malveillant [75].

- Taux de faux negatifs (FNR)

C'est la probabilité que l'IDS ne déclenche pas une alarme en cas d'intrusion. Cela a été mesuré par le pourcentage de FNR. Il ne déclenche pas de fausses alarmes négatives et ne laisse pas passer le trafic réseau lorsque les IDS ne disposent pas de la règle correspondante définie pour le trafic réseau [75].

- Taux de vraie Positive (TPR)

C'est la probabilité que l'IDS déclenche une alarme lorsqu'une intrusion est détectée. Cela a été mesuré par le pourcentage de TPR. Cela déclenche une véritable alarme positive lorsqu'ils détectent avec précision le trafic malveillant [76].

L'expérience a testé la précision de Snort et de Suricata dans deux conditions de test.

1. Mesurer les taux de faux positifs, de faux négatifs et de vrais positifs des deux IDS avec un trafic légitime.
2. Mesure les taux de faux positifs, de faux négatifs et de vrais positifs de deux IDS combinant un trafic légitime et un trafic malveillant à une vitesse de réseau fixe de 10 Gbps.

Un ensemble de règles par défaut et une configuration de performances par défaut des deux IDS ont été utilisés. Snort et Suricata ont des ensembles de règles identiques pour identifier le trafic réseau malveillant. Les différents types de trafic réseau malveillant ont été injectés aux deux IDS pour simuler les attaques. Enfin, chaque IDS inspectera le trafic légitime et malveillant et déclenchera des alarmes lorsque le trafic d'entrée correspond à l'ensemble de règles. Le nombre d'alarmes (faux positif, faux négatif et vrai positif) indiquera avec quelle précision Snort et Suricata classeront le trafic réseau. Le framework Metasploit est utilisé pour générer du trafic malveillant avec différents exploits et charges utiles.

Dans le domaine d'étude de la performance d'IDS, on s'intéresse à comparer les activités opérées avec les réactions de l'IDS. On retrouve quatre paramètres principaux :

- Vrai négatif ou True Negative (TN) : activité légitime reconnue comme telle.
- Vrai positif ou True Positive (TP) : activité malveillante reconnue comme telle.
- Faux positif ou False Positive (FP) : activité légitime reconnue comme une activité malveillante.
- Faux négatif ou False Négative (FN) : activité malveillante reconnue comme une activité légitime

À partir de ces paramètres, plusieurs autres peuvent être déduites :

- Taux de vraie positifs détection (TPR) : rapport entre le nombre de vrais positifs et le nombre total de vrais positifs et faux négatifs.

$$TPR = TP/(TP + FN) \quad [74]$$

- Taux de faux positifs (FPR) : rapport entre le nombre de faux positifs et le nombre total de faux positifs et vrais négatifs.

$$FPR = FP/(FP + TN) \quad [74]$$

– Taux de Faux négatif (FNR) : rapport entre le nombre de faux positifs et le nombre total de vrais positifs et faux négatif.

$$FNR = FN / (FN + TP) \quad [74]$$

Dans cette expérience ona analysé l'exactitude de la détection de Snort et de Suricata lors du traitement du trafic réseau légitime et malveillant. Les deux IDS ont été conservés au réglage par défaut. Le premier test de précision a été effectué à l'aide du générateur de trafic réseau légitime, qui injectait des paquets UDP, TCP et ICMP dans les deux IDS. Les fichiers journaux de Snort et Suricata ont été analysés et les résultats sont présentés dans la figure 59. Le taux de faux positifs (FPR) de Suricata était plus élevé lors du traitement de paquets UDP, TCP et ICMP que le FPR de Snort. Cependant, Snort n'a pas déclenché d'alarmes de taux réel positif (TPR) et de taux de faux négatif (FNR). Par rapport à Suricata, cela a entraîné un FPR environ de 40% et un TPR environ de 4%. Par conséquent, Snort a déclenché moins d'alarmes fausses positives. Alors que de fausses alarmes négatives ont été observées dans les deux IDS, la précision de détection de Snort s'est avérée supérieure à Suricata dans ce scénario.

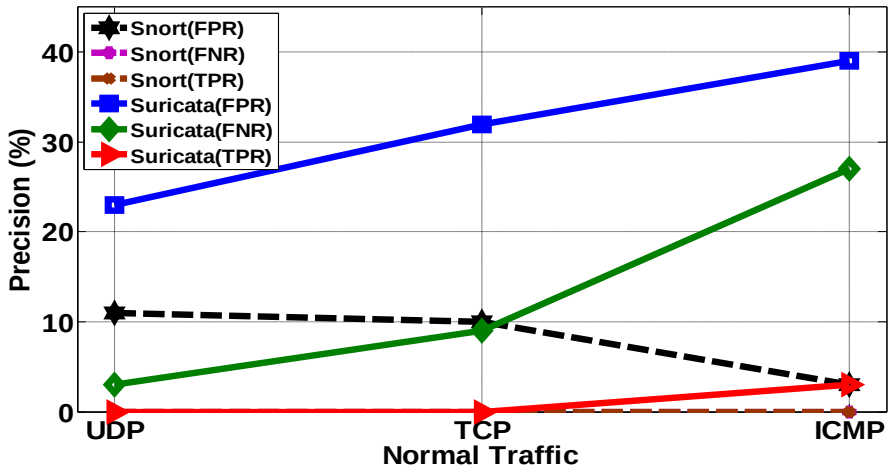

Figure 59 : Mesures de précision du trafic normal

La précision d'alarme réelle positive des deux IDS était à un taux acceptable et les deux IDS étaient configurés avec le jeu de règles par défaut. Le deuxième test a été réalisé avec un trafic combiné légitime et malveillant à une vitesse de réseau de 10 Gbps (Figure 60).

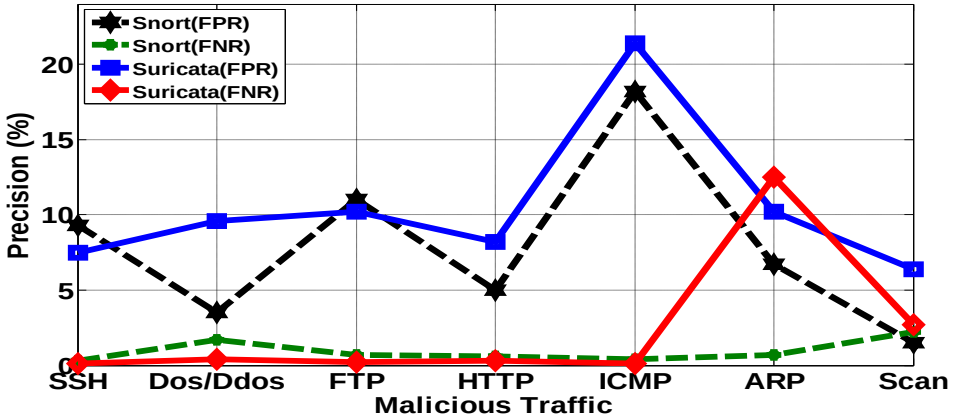

Figure 60 : Le taux de précision de trafic malveillant à 10 Gbps indiquant la moyenne

Avec les deux IDS configurés avec le réglage par défaut, Suricata a déclenché un taux élevé de fausses alarmes négatives. Lors de l'analyse du fichier journal de Suricata, il a été constaté que

l'ensemble de règles de Suricata ne traitait pas les trames de couche de liaison de données et qu'il était donc incapable de détecter ces trames malveillantes, et que la majorité des fausses alarmes négatives étaient déclenchées lorsque Suricata traitait les trames malveillantes ARP. Comparativement, la précision de Snort était de loin supérieure car elle déclenchait moins de fausses alarmes négatives.

Selon les résultats obtenus on trouve que Snort est un IDS plus précis que Suricata (Figure 60). À une vitesse supérieure de 10 Gbps, Snort supprime plus de paquets que Suricata, mais à des vitesses inférieures, le taux de perte de paquets est réduit. Le taux de traitement des paquets de Suricata s'avère meilleur que celui de Snort. Snort a une utilisation moindre du processeur et de la mémoire par rapport à Suricata pour différentes vitesses de trafic. Un problème avec Snort est son taux d'alarme faux positif, bien que meilleur que celui de Suricata. Il est crucial que cela soit résolu car l'analyse des fausses alarmes positives consomme du temps administrateur et des ressources de calcul.

6.6 Conclusion

Ce chapitre présente une évaluation comparative de deux systèmes, open-source, de détection d'intrusion. Snort est actuellement capable d'effectuer une analyse de trafic en temps réel et une journalisation de paquets sur des réseaux IP (Internet Protocol).Suricata, en est encore à ses débuts, mais offre des améliorations de la vitesse, de la maturité et des fonctionnalités non disponibles dans Snort. Suricata et Snort sont tous des systèmes de détection d'intrusion très performants, dotés chacun de forces et de faiblesses. Nous avons réalisé des expérimentations sur Suricata et Snort basées sur des données similaires pour tirer des recommandations. Snort et Suricata ont été évalués sur des machines virtuelles avec différents protocoles. Nous avons montré que les deux systèmes ont des faux positifs et des faux négatifs.

Nous avons montré que l'architecture de Suricata nécessite plus de mémoire et de ressources processeur (CPU) que Snort. L'utilisation globale du processeur de Suricata est supérieure à celle de Snort. Nous avons vu aussi que Suricata utilise plus de la quantité de RAM en comparaison avec Snort. Cela pourrait être attribué à la surcharge nécessaire à la gestion des multiples threads de détection dans Suricata. Suricata présente l'avantage de pouvoir évoluer pour prendre en charge un trafic réseau accru sans nécessiter plusieurs instances. Snort est léger et rapide, mais sa capacité à de traiter un grand flux de données reste limitée. Bien que les frais de traitement de Snort soient inférieurs à ceux de Suricata, la nécessité de multiples instances pour accomplir tout ce que Suricata peut réaliser avec sa conception multithread augmente les coûts d'exploitation et de gestion d'un environnement Snort. Les mesures de précision du trafic réalisées montrent que Suricata est un système performent par rapport à Snort. Ce qui lui permet d'être utilisé pour améliorer le système Snort existant.

Dans nos futurs travaux, nous allons étendre ce travail pour objectif d'étudier d'autres systèmes de détection d'intrusion et à d'autres approches hybrides possibles pour l'apprentissage automatique et le réglage précis des paramètres.

Conclusion et perspectives

Le système d'information d'une entreprise est un élément vital à son fonctionnement. Il est donc nécessaire d'assurer sa protection, afin de lutter contre les menaces qui pèsent sur l'intégrité, la confidentialité et la disponibilité des ressources.

Aujourd'hui, la sécurité du système d'information est quasi-indispensable pour le bon fonctionnement d'une entreprise. Aucune entreprise ne peut prétendre vouloir mettre en place un système d'information, quelque soit sa taille, sans envisager une politique de sécurité.

Dans ce travail nous avons abordé le problème de la sécurité des systèmes d'information, aussi nous avons proposé quelques solutions de contrôle d'accès, de filtrage des paquets et de détection d'intrusions.

Dans ce rapport, nous avons commencé par la description du principe, des objectifs, des domaines, et différentes facettes de la sécurité du système informatique. Puis nous avons expliqué les méthodes et les outils d'attaques informatiques. Pour savoir prendre les précautions nécessaires afin de stopper ou de limiter le nombre des menaces. Par la suite, nous avons donné un aperçu sur les mesures de sécurité, par l'implémentation des algorithmes de cryptographie afin de sécuriser le système d'information, et par l'utilisation des protocoles de sécurité afin de sécuriser la circulation de l'information confidentielle sur le support de communication, parmi les protocoles proposés dans cette thèse:

- Le protocole SSH qui est très simple de mise en place, permet d'effectuer sous un tunnel sécurisé des connexions à distance, des transferts de fichiers et d'autres fonctionnalités.
- Le protocole SSL/TLS est actuellement le protocole d'authentification et de sécurisation des échanges le plus déployé sur le web.
- Le protocole IPSEC est un protocole de sécurité au niveau réseau, permet la protection contre le contournement et l'analyse du trafic.

En outre, nous avons décrit les systèmes de détection d'intrusion (IDS) et nous avons expliqué les diverses méthodes de détection d'intrusion. Elles se basent principalement sur deux approches de détection : approche par scénario et approche comportementale. Dans ce travail nous avons implémenté le système de détection d'intrusion réseaux (NIDS) libre «SNORT », il est basé sur l'approche par scénario qui utilise une base de signature des attaques avec des outils d'interfaçage graphique afin de permettre à l'administrateur de redéfinir les règles de filtrage de trafic entrant et sortant du système informatique.

Dans le quatrième chapitre de cette thèse, nous avons décrit la virtualisation, les domaines d'application, les avantages offerts par cette technologie et les différents types de la virtualisation. Ensuite, nous avons cité les différentes solutions de virtualisation existantes et qui utilisent des technologies variées, en fonction des buts du projet. Enfin, nous avons proposé une approche de sécurisation d'une plateforme virtuelle basée sur la technologie firewalling et des systèmes de détection d'intrusions et prévention open source et des outils de visualisation graphiques en temps réel pour accélérer l'intervention s'il y on des intrusions. En guise de conclusion, nous pouvons affirmer que la mise en place des serveurs virtuels est possible au sein de n'importe quelle architecture.

Nous avons terminé ce rapport par une évaluation comparative des performances et des précisions de deux IDS open source Snort et Suricata

Dans le futur proche nous essayons d'implémenter le système de détection d'intrusion hybride et de réaliser des analyses comportementales du réseau permettant de définir les activités suspectes.

En ce qui concerne la mise en place des règles de sécurité au niveau de firewall virtuel, nous comptons travailler encore pour appliquer les algorithmes de BIG DATA sur les logs de l'infrastructure virtuelle afin de détecter les incidents en temps réel pour prendre la décision convenable immédiatement.

Publications

[1] Abdelkarim BEN CHARKE, Mohamed CHABI Mohamed Fakir, Contributionto the security of the information system, TELKOMNIKA Indonesian Journal of Electrical Engineering Vol 16, No 1 Octobre 2015

[2] Abdelkarim BEN CHARKE, Mohamed CHABI Mohamed Fakir, Comparative Performance Evaluation of Intrusion Detection System: Suricata and Snort, Bulletin of Electrical Engineering and Informatics, BEEI (Accepté pour la publication 2019)

Communications

[1] Abdelkarim BEN CHARKE, Mohamed FAKIR, Frequent Itemset Mining by Eclat, and SSDM alogirthms, First International Conference on Business Intelligence (CBI'14), Beni Mellal, Morocco April 29-30, 2014

[2] Abdelkarim BEN CHARKE, Mohamed CHABI,Mohamed FAKIR, Contributionto the security of the information system, Second International Conference on Business Intelligence (CBI'15), Beni Mellal, Morocco April 23-25, 2015

[3] Abdelkarim BEN CHARKE, Mohamed CHABI,Mohamed FAKIR, Securing the virtual KVM platform with Sophos UTM firewall,17th International Arab Conference on Information Technology (ACIT'2016), Beni Mellal, Morocco December 6-8, 2016

[4] Abdelkarim BEN CHARKE, Mohamed CHABI,Mohamed FAKIR, IPS open source for virtual infrastructure monitoring, Third International Conference on Business Intelligence (CBI'17), Beni Mellal, Morocco March 29-31, 2017

[5] Abdelkarim BEN CHARKE, Mohamed CHABI,Mohamed FAKIR, Mohamed BASLAM, Measurement of the security for an information system , 4th International Conference on Business Intelligence (CBI'17), Beni Mellal, Morocco April 25-27, 2018

Références

[1] Laurent Bloch, Christophe Wolfhugel, Nat Makarévitch, Sécurité informatique Principes et méthode à l'usage des DSI, RSSI et administrateurs, Eyrolles, 4e édition 2013.

[2] Jean-Marc Royer, Sécuriser l'informatique de l'entreprise : enjeux, menaces, prévention et parades, Editions ENI, 2004.

[3] Williams Stallings, Cryptography and Network Security Principles and Practice, Pearson, Sixth Edition 2013.

[4] Santanu Sarkar. Proving empirical key-correlations in RC4. Information Processing Letters, 2014, 114(5): 234-238.

[5] P. Karthigai Kumar, K. Baskaran, An ASIC implementation of low power and high throughput blowfish crypto algorithm, Microelectronics Journal 41 (2010) 347–355.

[6] Natassya B.F. Silva, Daniel F. Pigatto, Paulo S. Martins, Kalinka R.L.J.C Branco, Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer, Journal of Network and Computer Applications 60 (2016) 130-143.

[7] Manju Suresh, Neema M. Hardware implementation of blowfish algorithm for the secure data transmission in Internet of Things, Procedia Technology 25 (2016) 248 – 255.

[8] Balram Swami, Ravindar Singh, Sanjay Choudhary, Dual Modulus RSA based on Jordan-Totient function, Procedia Technology 24 (2016) 1581 – 1586.

[9] Van Quang Dao, Contribution à l'étude de la qualité de service pour les protocols sécurisés de télécommunication. Application à IPSEC, Thèse 12 décembre 2005.

[10] Cédric Liorens, Laurent Levier, Denis Valois, Tableaux de bord de la sécurité réseau, Eyrolles 3e édition 2010, Livre.

[11] Imtiaz Ahmad, A. Shoba Das, Hardware implementation analysis of SHA-256 and SHA-512 algorithms on FPGAs, Computers and Electrical Engineering 31 (2005) 345–360.

[12] Pierre-Alain Fouque, Gaëtan Leurent, Phong Q Nguyen. Full Key-Recovery Attacks on HMAC/NMACMD4 and NMAC-MD5. LNCS. 2007; 4622: 13-30.

[13] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[14] Avi Turiel, Commtouch. IPv6: new technology, new threats. Network security. 2011; 2011: 13-15.

[15] Kenneth G Paterson. A cryptographic tour of the IPsec standards. Information security, information security technical report 11 (2006) 72–81.

[16] Rafeeq Ur Rehman, Intrusion Detection Systems with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID, Prentice Hall; 01 (25 mai 2007), Livre

[17] Babaci Nabil, Recherche et exploitation de vulnérabilités des IDS par l'approche polymorphique, thèse, 23 juin 2012.

[18] Chirag N. Modi, Dhiren R. Patel, Avi Patel, Muttukrishnan Rajarajan,Integrating Signature Apriori based Network Intrusion Detection System (NIDS) in Cloud Computing, Procedia Technology 6 (2012) 905 – 912.

[19] Ali KARTIT, Une nouvelle approche de détection d'intrusions et étude des problèmes liés au déploiement de politiques de sécurité dans les réseaux informatiques, Thèse 05 Novembre 2011.

[20] Toby Kohlenberg, Raven Alder, Dr. Everett F. (Skip) Carter, Jr, James C. Foster, Matt Jonkman, Raffael Marty, Eric Seagren, Snort® IDS and IPS Toolkit, Syngeress Network, 2007, Livre.
[21] Al-Sakib Khan Pathan, The State of the Art in Intrusion Prevention and Detection, CRC Press, 29 jan 2014, Livre.
[22] K Salah, A Kahtani. Performance evaluation comparison of Snort NIDS under Linux and Windows Server. Journal of Network and Computer Applications. 2010, 33(1) : 6-15.
[23] Jack Koziol, Intrusion Detection with Snort, Pearson Education (US) 2003.
[24] Michaël AMAND,Mohamed NSIRI, Etude d'un système de détection d'intrusion comportemental pour l'analyse du trafic aéroportuaire, IENAC 08 27 janvier 2011.
[25] Ibrahim HAJJEH « Sécurité des échanges : Conception et validation d'un nouveau protocole pour la sécurisation des échanges » Thèse, 7 décembre 2004.
[26] Pawel Skrobanek, Intrusion Detection Systems, IntechOpen, Mars 2011.
[27] EC-Council, Network Defense: Security Policy and Threats, Cengage Learning (1754) 2010.
[28] Xavier Cacelle, Réseaux CPL par la pratique, Eyrolles 2006, Livre.
[29] Solange Ghernaouti, Sécurité informatique et réseaux - 4e édition : Cours avec plus de 100 exercices corrigés, Dunod 2013 , livre.
[30] Fernando Rodr´ıguez-Haro, Felix Freitag, Leandro Navarrob, Efrain Hernandez-Sanchez, Nicandro Farias-Mendoza, Juan Antonio Guerrero-Ibanez, Apolinar Gonzalez-Potes, A summary of virtualization techniques, Procedia Technology 3 (2012) 267 – 272.
[31] David E. Williams, Virtualization with Xen(tm): Including XenEnterprise, XenServer, and XenExpress, Syngress (3 juillet 2007), Livre.
[32] Chris Takemura, Luke S. Crawford, The Book of Xen: A Practical Guide for the System Administrator, No Starch Press, 2009 Livre.
[33] Damien BRULEY, René-François MENNECIER, Eric MAILLÉ, VMware - Les solutions de virtualisation au sein de votre organisation (serveur et poste de travail), Editions ENI 2012, Livre.
[34] Marisol García-Valls, Tommaso Cucinotta, Chenyang Lu, Challenges in real-time virtualization and predictable cloud computing, Journal of Systems Architecture 60 (2014) 726–740.
[35] Shing-Han Li, David C. Yen, Shih-Chih Chen, Patrick S. Chen, Wen-Hui Lu, Chien-Chuan Cho, Effects of virtualization on information security, Computer Standards & Interfaces 42 (2015) 1–8.
[36] Feng Wang, Xiaoyang Sun, Shi Li, Yong Wang, Bingjia Xiao, Sidi Chang,The implementation of virtualization technology in EAST data system, Fusion Engineering and Design 89 (2014) 766–769.
[37] Simon Grinberg, Shlomo Weiss, Architectural virtualization extensions: A systems perspective, Computer Science Review 6 (2012) 209–224.
[38] Gal Motika, Shlomo Weiss, Virtio network paravirtualization driver: Implementation and performance of a de-facto standard, Computer Standards & Interfaces 34 (2012) 36–47.
[39] Mahdi Daghmehchi Firoozjaei, Jaehoon (Paul) Jeong, Hoon Ko, Hyoungshick Kim, Security challenges with network functions virtualization, Future Generation Computer Systems 67 (2017) 315–324.
[40] Sreejith.C,VMware NSX Network Essentials, Packt Publishing (30 septembre 2016), Livre.

[41] N.M. Mosharaf Kabir Chowdhury, Raouf Boutaba. A survey of network virtualization, Computer Networks 54 (2010) 862–876.
[42] Vaezi, Zhang. Y., Chapiter 2: Virtualization and Cloud Computing, Cloud Mobile Networks, 2017.
[43] Philippe Gillet, Virtualisation des systèmes d'information avec Vmware - Architecture, projet, sécurité et retours d'expérience, Editions ENI (17 août 2009), Livre.
[44] Yaoxue Zhang, Yuezhi Zhou, Separating computation and storage with storage virtualization, Computer Communications 34, 2011.
[45] Si Zhen-yu. Storage System Design Scheme in Virtualization Construction. Journal of Northeast Agricultural University, Dec. 2012.
[46] Eric MAILLÉ, VMware vSphere 4 - Mise en place d'une infrastructure virtuelle, Editions ENI (11 janvier 2010), Livre.
[47] Edwar Ali, Susandri, Rahmaddeni, Optimizing Server Resource by Using Virtualization Technology, Procedia Computer Science 59 (2015) 320 – 325.
[48] Qian Lin, Zhengwei Qi, Jiewei Wu, Yaozu Dong, Haibing Guan, Optimizing virtual machines using hybrid virtualization, The Journal of Systems and Software 85 (2012) 2593–2603.
[49] Antoine Capra, Virtualisation en contexte HPC, Thèse, Décembre 2015.
[50] Rick F. van der Lans , Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouse, Elsevier Science & Technology (30 août 2012).
[51] Benoit Morgan, Protection des systèmes informatiques vis-à-vis des malveillances : un hyperviseur de sécurité assiste par le matériel, Thèse, Décembre 2016.
[52] Luigi Coppolino, Salvatore D'Antonio, Giovanni Mazzeo, Luigi Romano, Cloud security: Emerging threats and current solutions, Computers and Electrical Engineering V59 (2017) 126–140.
[53] Akinbi A, Pereira E, Beaumont C. Evaluating security mechanisms implemented on public platform-as-a-service cloud environments case study: Win- dows azure. In: Internet technology and secured transactions (ICITST), 2013 8th international conference for, London; 2013. p. 162–7. doi: 10.1109/ ICITST.2013.6750183.
[54] Minhaj Ahmad Khan, A survey of security issues for cloud computing, Journal of Network and Computer Applications 71(2016)11–29.
[55] Luigi Coppolino, Salvatore D'Antonio, Giovanni Mazzeo, Luigi Romano, Cloud security: Emerging threats and current solutions, Computers and Electrical Engineering 59 (2017) 126–140.
[56] Wenke Lee, Cliff Wang, David Dagon, Botnet Detection: Countering the Largest Security Threat, Springer, 2008.
[57] Yinqian Zhang, Ari Juels, Michael K. Reiter, Thomas Ristenpart, Cross-VM Side Channels and Their Use to Extract Private Keys, ACM, 2012
[58] Shillpi Chandna, Rohit Singh, Fazil Akhtar, Data scavenging threat in cloud computing, International Journal of Advances In Computer Science and Cloud Computing Volume 2, Nov 2014.
[59] Joaquin Garcia -Alfaro, Frédéric Cuppens , Nora Cuppens-Boulahia, Salvador Martinez, Jordi Cabot, Management of stateful firewall misconfiguration, Computers & Security 39 (2013) 64-85.
[60] Ric Messier, Network Forensics, John Wiley & Sons Inc (28 juillet 2017), Livre.

[61] Detection of Intrusions and Malware, and Vulnerability Assessment: 11th International Conference, DIMVA 2014 Proceedings.
[62] David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni, Hacking sécurité et tests d'intrusion avec Metasploit, Pearson, 30 août 2013.
[63] Ali A. Ghorbani, Wei Lu, Mahbod Tavallaee, Network Intrusion Détection and Prévention, Springer, 2010.
[64] Thibaut Probst, Évaluation et analyse des mécanismes de sécurité des réseaux dans les infrastructures virtuelles de cloud, Thèse, Octobre 2016.
[65] Alex Shenfield, David Day, Aladdin Ayesh, Intelligent intrusion detection systems using artificial neural networks, ICT Express 4, Avril 2018.
[66] Kirill Shipulin, Positive Technology, We need to talk about IDS signatures, Network Security, March 2018.
[67] K Salah, AKahtani. Performance evaluation comparison of Snort NIDS under Linux and Windows Server. Journal of Network and Computer Applications, 2010, 33(1): 6-15.
[68] Colin Tankard, New rules for combating new threats, Computer Fraud & Security, Volume 2014, Issue 4, April 2014, Pages 14-16.
[69] Mohamed M. Abd-Eldayem, A proposed HTTP service based IDS, Egyptian Informatics Journal (2014) 15, 13–24.
[70] Stephen Northcutt, Judy Novak, Network Intrusion Detection, third ed. New RidersPublishing, Indianapolis, 2003 pp. P79, P401–404.
[71] Bruno BogazZarpelão, , Rodrigo SanchesMianib , Cláudio Toshio Kawakania, Sean Carlisto de Alvarenga, A survey of intrusion detection in Internet of Things, Journal of Network and Computer Applications 84 (2017) 25–37.
[72] SolaneDuquea, Dr.Mohd. Nizam bin Omar, Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS), Procedia Computer Science 61 (2015) 46–51.
[73] Martin Grill, Tomáš Pevný, Martin Rehak, Reducing false positives of network anomaly detection by local adaptive multivariate smoothing, Journal of Computer and System Sciences 83 (2017) 43–57.
[74] Mehrnaz Mazini, Babak Shirazi, Iraj Mahdavi Anomaly network-based intrusion detection system using a reliable, Journal of King Saud University – Computer and Information Sciences, 27 March 2018.
[75] Adeeb Alhomoud, Rashid Munir, Jules PagnaDisso, Irfan Awan, A. Al-Dhelaan, Performance Evaluation Study of Intrusion Detection Systems, Procedia Computer Science 5 (2011) 173–180.
[76] Kittikhun Thongkanchorn, Sudsanguan Ngamsuriyaroj, Vasaka Visoottiviseth, Evaluation studies of three intrusion detection systems under various attacks and rule sets 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013).