République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Abderrahmane Mira de Béjaïa

Faculté des sciences exactes

Département d'informatique

Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master en informatique

Spécialité : Administration et sécurité des réseaux(ASR)

Thème

Détection d'intrusions dans les réseaux LAN : IDS Snort sous LINUX

Réalisé par :

M. ABBAS Massinissa

M. AOUADI Djamel

Soutenu devant le jury composé de :

Président : M. Moktfi Mohand	Université de Béjaïa
Examinateur : M. Kadjouh Nabil	Université de Béjaïa
Examinateur : M. Bouchebah Fatah	Université de Béjaïa
Promoteur : M. Chekrid Mohamed	Université de Béjaïa

Promotion 2016/2017

Remerciements

Nos remerciements vont à Dieu le tout puissant qui nous a donné le courage et la volonté pour réaliser ce modeste travail.

Nous tenons aussi à remercier notre promoteur Mr CHEKRID Mohamed *pour ses précieux conseils et son orientation.*

Nos remerciements sont adressés également é nos chers parents pour leurs soutien et sacrifices consentis è notre égard.

A toute personne ayant contribué de prés ou de loin à la réussite de ce travail.

Nous tenons à remercier les membres de jury pour nous avoir fait l'honneur d'examiner et d'évaluer notre modeste travail.

Dédicaces

A nos chers parents auxquels nous devons toutes nos reconnaissances.

A nos frères et sœurs.

A nos familles sans exception.

A tous nos chers amis.

A tous ceux qui nous ont apporté de l'aide.

TABLE DES MATIERES

Tables des matieres

Table des matières……i

Liste des figures……v

Liste des tableaux……vii

Liste des abréviations……viii

Introduction générale

Chapitre I : Généralités sur la sécurité informatique

Introduction……..1

I.1. Sécurité informatique……..1

I.1.1. Définition……..1

I.1.2. Objectifs de la sécurité informatique……..1

I.1.3. Problèmes liés à la sécurité informatique……..2

I.1.4. Les attaques informatiques……..3

I.1.4.1. Définition……..3

I.1.4.2. Classification des attaques……3

I.1.4.3. les types d'attaques……4

I.1.4.4. Quelques attaques courantes……..6

I.1.5. Politique de Sécurité……..7

I.1.6. Mécanismes de sécurité des réseaux……7

I.2. Réseaux informatiques……..11

I.2.1. La norme OSI……..11

I.2.1.1. Définition……11

I.2.1.2.Les couches du modèle OSI……12

I.2.2. Le model TCP/IP……13

I.2.2.1. Définition……..13

Conclusion……14

Chapitre II : Systèmes de détection et de prévention d'intrusions

Introduction……..15

TABLE DES MATIERES

II.1. Détection d'attaque……..……..15
II.1.1. Système de détection d'intrusions(IDS)…….....……15
II.1.2. Principe de fonctionnement d'un IDS……16
II.1.3. Classification des IDS……17
II.1.4. Les types d'IDS……..……18
II.1.5. Méthodes de détection d'intrusion……21
II.1.6. Méthodes d'analyse……...……23
II.1.7. Les fonctions proposées par les IDS…….…24
II.1.8. Efficacité des IDS……..24
II.2. Systèmes de prévention d'intrusion(IPS)……..……24
II.2.1. Les systèmes de prévention d'intrusions réseau NIPS……..……25
II.2.2. Les systèmes de prévention d'intrusion du Hote (HIPS)…….……..25
II.2.3 Architecture d'un IPS…….…26
II.2.3. Comparaison entre IDS et IPS…….……..27
Conclusion……...….28
Chapitre III : Etat de l'art
Introduction……...……29
III.1. Haystack……...…29
III.1.1. Structure conceptuelle de Haystack…….….29
III.2. MIDAS (Multi Intrusion Detection and Alerting System)……..……30
III.3. IDES (Intrusion Detection Expert System)……..31
III.3.1. Structure de l'IDES……...……32
III.4. GrIDS……...……33
III.4.1. Architecture de GrIDS……..34
Conclusion……34
Chapitre IV : Mise en œuvre
Introduction……...……35
IV.1. Présentation de SNORT……..……35
IV.1.1. Fonctionnement de SNORT……..35
IV.1.2. Positionnement de SNORT dans le réseau……...35
IV.1.3. Architecture de SNORT……36
IV.1.4. Environnement……..37
IV.2. Paramétrage de SNORT……...…37
IV.2.1. Préprocesseurs…….…….37

TABLE DES MATIERES

IV.2.1.1. Mainfrag……..……..37
IV.2.1.2.HTTP decode…….....…….38
IV.2.1.3. Détecteur du balayage du port(Portscan Detector)……38
IV.2.1.4. Défragmentation(Defrag)……...……38
IV.2.1.5. Stream.……....38
IV.2.2. Les plugins de sortie……....39
IV.2.2.1. Alerte syslog (Alert_syslog)…….……39
IV.2.2.2Alerte rapide (alert_fast)…….…39
IV.2.2.3. Alerte pleine (Alert_full)……...39
IV.2.2.4. Alerte smb…….…39
IV.2.2.5. Alerte unixsock……..40
IV.2.2.6. Log_tcpdump……40
IV.2.3. Les bases de Snort……40
IV.2.3.1. Les inclusions…….. 40
IV.2.3.2. Les variables……..…40
IV.2.4. Les règles de Snort……..……...41
IV.2.4.1. Création des règles :……..……41
IV.2.4.1.1. Header……...……41
IV.2.4.1.2. Options……41
IV.3. Barnyard2……42
IV.3.1. Le plugin « unified2 »……...……42
IV.4. La console B.A.S.E……..……43
IV.5. Installation de Snort…….……43
IV.5.1. L'installation de l'outil Snort……43
IV.5.2.L'installation des regles Snort……..……43
IV.6. Lancement de snort…….……44
IV.6.1. Les modes de fonctionnement……..44
IV.6.1.1. Le mode ecoute (sniffer mode)…….…45
IV.6.1.2. Le mode NIDS……..….46
IV.7. Mise en place de Barnyard2……..…48
IV.7.1. Installation de Barnyard2……..48
IV.7.2. Configuration du fichier barnyard2.conf ……..…...49

TABLE DES MATIERES

IV.8. Mise en place de la base de données MySQL…….……51
IV.8.1. Installation.……51
IV.8.2. Création de la base de données pour Snort……..…….52
VI.8.3. Installation de snortmysql……....52
IV.9. Mise en place de la console B.A.S.E……...……54
IV.9.1. Installation des pré-requis……....54
IV.9.2. Configuration du fichier php.ini……..…….54
IV.9.3. Installation de B.A.S.E……...……..54
IV.9.4. Installation d'Adodb……...……..54
IV.10. Lancement d'attaque……..……58
Conclusion.……61
Conclusion générale
Bibliographie

Liste des figures

Figures I.1 : Attaque par interruption……...……3

Figures I.2 : Attaque par interception....…….…….3

Figures I.3 : Attaque par modification……...…4

Figures I.4 : Attaque par fabrication……..4

Figures I.5 : Attaque direct……...…….…..4

Figures I.6 : Attaque indirect par rebond……...…5

Figures I.7 : Attaque indirect par réponse……...……..6

Figures I.8 : Control du flux entrant…….…..8

Figures I.9 : Control du flux sortant……...……8

Figures I.10 : Position du NAT……...…….9

Figures I.11 : Emplacement d'une DMZ dans un réseau…….…9

Figures I.12 : Principe d'un proxy……10

Figures I.13: Emplacement d'un proxy dans un réseau……..10

Figures I.14 : Principe d'un VPN……..11

Figures I.15 : Le modèle OSI…….12

Figures I.16 : Les modèles TCP/IP et OSI…….…..13

Figures II.1 : Emplacement d'un IDS dans un réseau……15

Figures II.2 : Schéma d'architecture IDWG d'un IDS……...16

Figures II.3 : Classification des systèmes de détection d'intrusion……18

Figures II.4 : Déploiement du NIDS…….19

Figures II.5 : Déploiement du HIDS…….....20

Figures II.6 : Architecture d'un IPS…….26

Figure III.1 : Architecture conceptuelle de Haystack…….30

Figure III.2 : Structure de l'IDES…….32

LISTE DES FIGURES

Figure III.3 : Architecture de GrIDS……34

Figure IV.1 : Les différentes positions de SNORT dans un réseau……..… 36

Figure IV.2 : L'architecture de SNORT……..…….37

Figure IV.3 : Format des règles de Snort……..…41

Figure IV.4 : Vérification de l'installation de Snort……45

Figure IV.5 : Lancement du mode sniffer…….....45

Figure IV.6 : Format du paquet analyé en mode sniffer…….46

Figure IV.7 : Spécification de l'adresse réseau à surveiller……46

Figure IV.8 : Spécification du répertoire contenant les règles de Snort……...…….47

Figure IV.9 : Annulation de la méthode de sortie tcpdump……47

Figure IV.10 : Spécification de la méthode de sortie unified2……48

Figure IV.11 : Vérification de l'installation de Barnyard2…….49

Figure IV.12 : Configuration du fichier Barnyard2.conf……49

Figure IV.13 : Configuration de la sortie vers la base de données……...…….50

Figure IV.14 : Création du fichier Barnyard.waldo……..…..50

Figure IV.15 : Récupération du dernier timestamp……..…..50

Figure IV.16 : Injection du timestamp dans barnyard.waldo……...……51

Figure IV.17 : Lancement du service MySQL…….....51

Figure IV.18 : Création de la base de données……....52

Figure IV.19 : Vérification de la création de la base de données……...……....53

Figure IV.20 : Schématisation de la base de données…….53

Figure IV.21 : Lancement du service http……...…54

Figure IV.22 : Installation de la bibliothèque ADODB……..……55

Figure IV.23 : Page d'accueil de B.A.S.E……..……..55

Figure IV.24 : Sélection de la langue pour B.A.S.E……56

Figure IV.25 : Insertion des informatios relatives à la base de données……..56

Figure IV.26 : Insertion des coordonnées de l'administrateur…….…57

Figure IV.27 : Ajout des extentions de B.A.S.E……...57

Figure IV.28 : Résultat de l'installation de B.A.S.E.. ……...…..58

Figure IV.29 : Résultat de l'installation de B.A.S.E …….……..58

Figure IV.30 : Lancement d'une attaque sue le port TCP…….……59

Figure IV.31 : Détection d'une attaque sue le port TCP……...…….59

Figure IV.32 : Lancement d'une attaque sue le port UDP…….……60

Figure IV.33 : Détection d'une attaque sue le port UDP……....60

Liste des tableaux

Tableaux II.1 : Comparaison entre HIDS et HIPS……..27

Tableaux II.2 : Comparaison entre NIDS et NIPS……..28

Liste des abréviations

ACID: Analysis Console for Inrtusion Data Base

ADODB: Active Data Object Databae

DAQ : Data Acquisition

DHCP: Dynamic Host Configuration Protocol

DMZ: DeMilitarized Zone

DNS: Domain Name System

DOS: Denial Of Service

GCC : GNU Compiler Collection

HIDS: Host Intrusion Detection System

HIPS: Host Intrusion Prevention System

HTTP: HyperText Transfer Protocol

ICMP: Internet Control Message Protocol

IDES: Intrusion Detection Expert System

IDS: Intrusion Detection System

IDWG: Intrusion Detection exchange format Working Group

IP: Internet Protocol

IPS: Intrusion Prevention System

LAN: Local Area Network

MIDAS: Multi Intrusion Detection and Alerting System

MITM: Man In The Middle

NAT: Network Address Translation

NFS: Network File System

NIDES : Network Intrusion Detection Expert System

LISTE DES ABREVIATION

NIDS: Network Intrusion Detection System

NIPS: Network Intrusion Prevention System

NTIC: Network Technologies of information and communication

OSI: Open Systems Interconnection

PHP: HypertextPreprocessor

SGBD: Système de Gestion de Base de Données

SQL: Structured Query Language

SI: Système d'information

TCP: Transmission Control Protocol

TCP/IP: Transmission Control Protocol/Internet Protocol

UDP: User Datagram Protocol

URL: Uniform Ressource Lactor

VPN: Virtual Private Network

Introduction générale

Avec l'évolution des technologies de l'information et des communications (NTIC), les systèmes d'information sont aujourd'hui de plus en plus ouverts sur le monde extérieur notamment Internet. Cette ouverture simplifie considérablement la vie pour l'homme en lui offrant plusieurs services, et ce à travers des centaines de millions d'ordinateurs reliés à internet. Cependant, cette interconnexion des ordinateurs permet également aux utilisateurs malveillants d'utiliser ces ressources à des fins abusives et de lancer par exemple des attaques de divers types à l'encontre des serveurs web.

La sécurité des systèmes informatique est une problématique d'une importance capitale pour les individus ainsi que pour les entreprises. Elle repose en premier lieu sur la mise en place d'une politique de sécurité autour de ces systèmes. Outre la mise en place de pare-feu et de systèmes d'authentification de plus en plus sécurisés, il est nécessaire, pour compléter cette politique de sécurité, d'avoir des outils de surveillance pour auditer le système d'information et détecter d'éventuelles intrusions.

Afin de compléter cette tâche d'une manière plus sûre et plus sécurisée, l'utilisation des mécanismes de détection d'intrusion s'imposent. Les systèmes de détection d'intrusions (intrusion détection system, IDS) ont été conçus pour surveiller les systèmes d'informations(SI) et découvrir les violations de la politique de sécurité automatiquement, et ainsi en informer l'opérateur.

C'est dans cette optique que s'inscrit notre mémoire, à savoir l'étude des systèmes de détection d'intrusion dans les réseaux locaux. Ce dernier est structuré en quatre chapitres comme suit :

Le premier chapitre est consacré à la présentation des concepts de base concernant les réseaux et de la sécurité informatique à savoir les menaces, les logiciels malveillants, les politiques et les mécanismes de sécurité.

Dans le second chapitre, nous présentons les systèmes de détection d'intrusion(IDS), à savoir leur type, leur mode de fonctionnement, leur classification et les méthodes de détection d'intrusions, etc.

Dans le troisième chapitre, nous donnons une liste d'IDS les plus importants existants dans la littérature, puis nous présentons certains d'eux de manière générale en donnant leur architecture et principe de fonctionnement.

Le quatrième chapitre sera consacré à l'étape de la mise en place de notre IDS. Cette étape comporte trois parties : présentation des différents outils utilisés, procédure d'installation de l'IDS Snort sous Linux, et mise en œuvre et évaluation.

Enfin, nous terminons le mémoire par une conclusion générale où des perspectives de recherche à explorer à l'avenir seront données.

Introduction

L'ouverture des systèmes d'informations sur Internet, engendré par les besoins multiples des utilisateurs et des réseaux d'entreprises, à donner lieu à plusieurs problèmes de sécurité, en particulier les attaques Informatiques. Le nombre croissant de ces attaques a amené le monde informatique à mettre en place des systèmes de sécurité afin de les contre.

Dans ce chapitre, nous présentons des notions de base de la sécurité informatique à savoir les principales menaces pesant sur la sécurité des réseaux, et quelques mécanismes de défense retenus pour faire face aux différentes menaces.

I.1. Sécurité Informatique

I.1.1. Définition

La sécurité informatique est l'ensemble des moyens techniques, juridiques, organisationnels et humains mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles. [1]

I.1.2. Objectifs de la sécurité informatique

L'objectif de la sécurité informatique est de garantir que les ressources matérielles et/ou logicielles d'un système sont utilisées dans un cadre prévu. Il convient de garantir les services suivants :

a) La confidentialité : elle consiste à s'assurer que les informations confidentielles ne sont accessibles que pour des personnes autorisées.

b) L'intégrité : elle consiste à garantir que les données protégées ne peuvent être modifiées que par les personne non autorisée.

c) La disponibilité : elle consiste à garantir l'accès à un service ou à une ressource pour les personnes autorisées.

d) La non-répudiation : il s'agit de s'assurer qu'un correspondant participant dans une transaction ne peut nier son implication.

e) L'authentification : elle permet de s'assurer de l'identité d'un utilisateur donné. C'est-à-dire garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être.

f) Le contrôle d'accès : il consiste à limiter et de contrôler l'accès à des systèmes et à des applications via des maillons de communication. Ainsi, chaque entité essayant d'obtenir un accès à une ressource doit d'abords être s'authentifier.

I.1.3. Problèmes liés à la sécurité informatique : Différents types de problèmes, de risques et de menaces liés à la sécurité des systèmes informatiques, les plus importants sont :

a) Vulnérabilités : une vulnérabilité est une faille ou une faiblesse dans le système qui peut être exploitée par une personne mal intentionné. Elle peut être humaine, technologique, organisationnelle ou une mise en œuvre.

b) Menaces : évènement pouvant nuire au système. On distingue deux catégories de menace :

- **Menaces passives** : permettent à une personne malveillante d'intercepter des informations dans un réseau. Ainsi, nuire à la confidentialité des données.
- **Menaces actives :** consistent à altérer des informations dans un réseau. Ainsi nuire au bon fonctionnement du réseau.

Intrusions : une intrusion est une attaque malveillante d'origine interne ou externe, qui permet à un utilisateur illégitime d'exploiter une vulnérabilité dans le système afin contrôler ou d'accéder à certaines ressources. Une intrusion peut prendre la forme d'un virus, d'un ver ou d'un cheval de Troie.

c) Logiciels malveillants :

Un logiciel malveillant ou malicieux (en anglais : malware), parfois logiciel nuisible est un programme développé dont le but est de nuire à un système informatique, sans le consentement de l'utilisateur dont l'ordinateur est infecté. Les logiciels malveillants peuvent être classés en fonction des trois mécanismes suivants :le mécanisme de propagation, le mécanisme de déclenchement ou le mécanisme de charge utile « *Wikipédia* ».

- **Les virus** : un virus est un programme qui s'exécute et se reproduit automatiquement. Il est capable de perturber le bon fonctionnement d'un ordinateur sans que l'utilisateur s'en aperçoive.
- **Les vers :** un ver est une variété de virus qui se reproduit et se propage dans un réseau à l'insu des utilisateurs.
- **Les chevaux de Troie** : un cheval de Troie est un logiciel en apparence légitime, mais une fois installé sur un ordinateur il effectue des actions cachées et pernicieuse.
- **Porte dérobée :** une porte dérobée est un logiciel de communication caché, installé par un virus ou par un cheval de Troie, qui donne à un agresseur extérieur l'accès à un ordinateur via un réseau.
- **Les logiciels espions :** un logiciel espion est un programme qui collecte, à l'insu des utilisateurs légitimes, des informations au sein du système où il est installé et les communique à un agent extérieur sans avoir obtenu au préalable une autorisation.
- **Le spam** : courrier électronique non sollicité, souvent publicitaire. Ils encombrent le réseau et font perdre du temps à leurs destinataires.

I.1.4. Les attaques informatiques

I.1.4.1. Définition : une attaque informatique est définie comme une faute d'interaction malveillante visant à violer une ou plusieurs propriétés de sécurité. C'est une faute externe créée avec l'intention de nuire, y compris les attaques lancées par des outils automatiques : vers, virus, etc [2].

I.1.4.2. Classification des attaques

On peut distinguer quatre catégories types d'attaques :

1) **Attaque par interruption :** Un atout du système est détruit ou devient indisponible ou inutilisable. C'est une attaque portée à la disponibilité. La destruction d'une pièce matérielle (disque dur), la coupure d'une ligne de communication, ou la mise hors service d'un système de gestion de fichiers en sont des exemples [3].

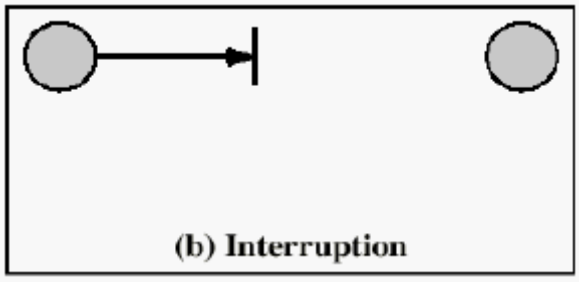

Figure I.1 : Attaque par interruption [3].

2) **Attaque par interception :** une tierce partie non autorisée obtient un accès à un atout. C'est une attaque portée à la confidentialité. Il peut s'agir d'une personne, d'un programme ou d'un ordinateur. Une écoute téléphonique dans le but de capturer des données sur un réseau, ou la copie non autorisée de fichiers ou de programmes en sont des exemples [3].

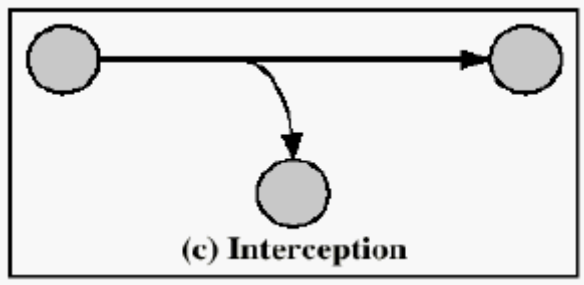

Figure I.2 : Attaque par interception [3].

3) **Attaque par modification :** Une tierce partie non autorisée obtient accès à un atout et le modifie de façon (presque) indétectable. Il s'agit d'une attaque portée à l'intégrité. Changer des valeurs dans un fichier de données, altérer un programme de façon à bouleverser son comportement ou modifier le contenu de messages transmis sur un réseau sont des exemples de telles attaques [3].

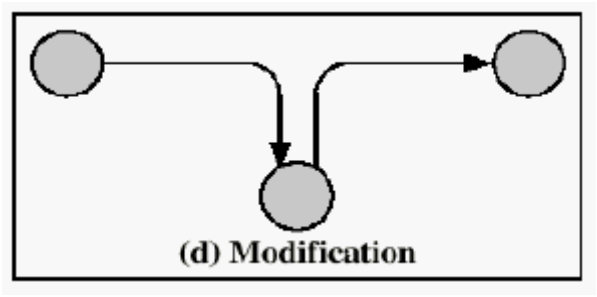

Figure I.3 : Attaque par modification [3].

4) **Attaque par fabrication :** Une tierce partie non autorisée insère des contrefaçons dans le système. C'est une attaque portée à l'authenticité. Il peut s'agir de l'insertion de faux messages dans un réseau ou l'ajout d'enregistrements à un fichier [3].

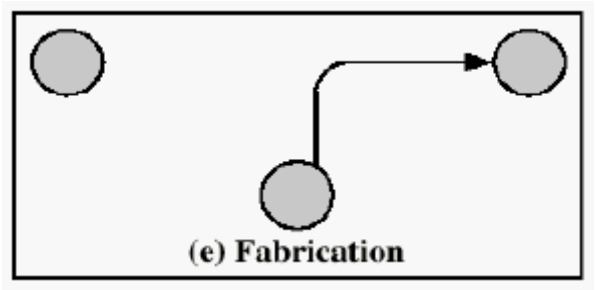

Figure I.4 : Attaque par fabrication [3].

I.1.4.3. Les Types d'attaques

Attaque directe : c'est la plus simple des attaques.

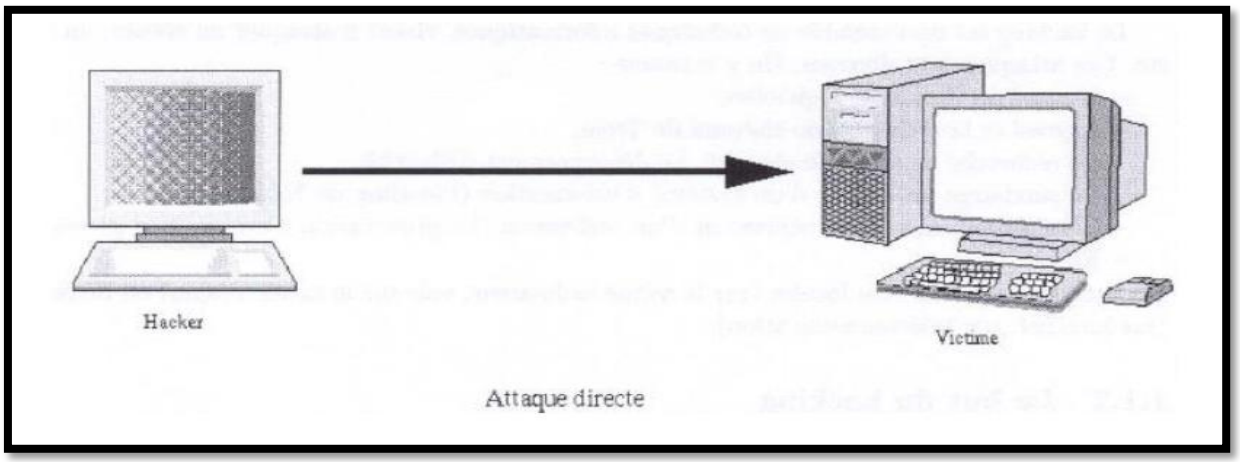

Figure I.5 : Attaque directe [4].

- Le hacker attaque directement sa victime à partir de son ordinateur pas des scripts d'attaque faiblement paramétrables.

- Les programmes de hack que les hackers utilisent envoient directement les paquets à la victime.
- Dans ce type d'attaque, il est possible de remonter à l'origine de l'attaque et identifier ainsi l'identité de l'attaquant [4].

➢ **Attaque indirecte par rebond**

Figure I.6 : Attaque indirecte par rebond [4].

Lors d'une attaque, les pirates gardent toujours à l'esprit le risque de se faire repérer, c'est la raison pour laquelle ils privilégient habituellement les attaques par rebond (par opposition aux attaques directes). Une attaque par rebond consiste à attaquer une machine par l'intermédiaire d'une autre machine afin de masquer toutes les traces permettant de remonter à lui (telle que son adresse IP) [4].

Attaque indirecte par réponse : Cette attaque est une dérivée de l'attaque par rebond.

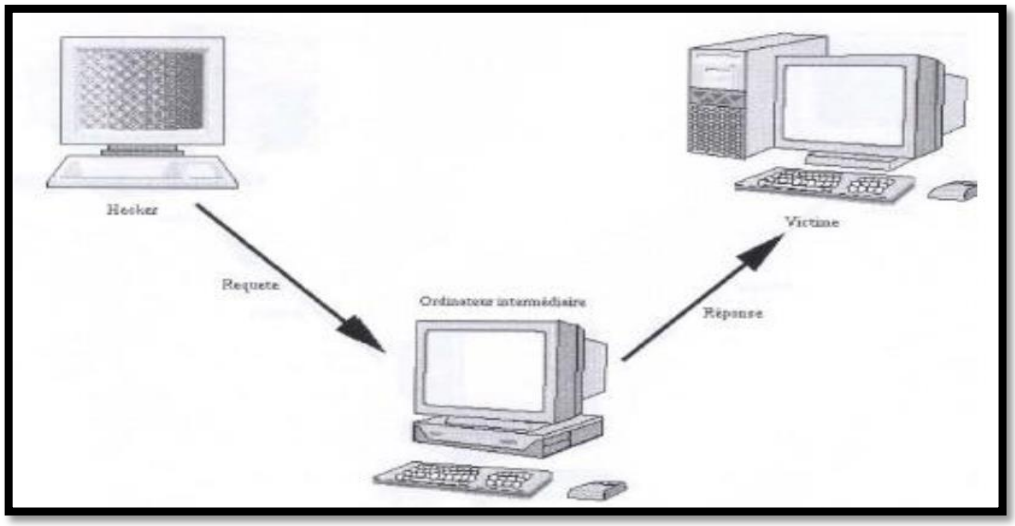

Figure I.7 : attaque indirecte par réponse [4].

- Elle offre les mêmes avantages, du point de vue du hacker.
- Mais au lieu d'envoyer une attaque à l'ordinateur intermédiaire pour qu'il la répercute, l'attaquant va lui envoyer une requête.
- Et c'est cette réponse à la requête qui va être envoyée à l'ordinateur victime [4].

I.1.4.4. Quelques attaques courantes

1. **Le déni de service**

Le but d'une telle attaque n'est pas de dérober des informations sur une machine distante, mais de paralyser un service ou un réseau complet ;ainsi les utilisateurs ne peuvent plus accéder aux ressources. Les deux exemples principaux sont le « ping flood » et l'envoi massif de courrier électronique pour saturer une boîte aux lettres *(mailbombing)*. La meilleure parade est le firewall ou la répartition des serveurs sur un réseau sécurisé. [5]

2. **IP spoofing**

Cette attaque consiste à se faire passer pour une autre machine en falsifiant son adresse IP. Elle est en fait assez complexe. Il existe des variantes, car on peut spoofer aussi des adresses email, des serveurs DNS ou NFS. [6]

3. **Le sniffing**

Cette attaque est utilisée par les pirates informatiques pour obtenir des mots de passe. Grâce à un logiciel appelé renifleur de paquets (sniffer), on peut intercepter toutes les paquets qui circulent sur un réseau. Par exemple, lors d'une connexion grâce à « telnet » le mot de passe de l'utilisateur va transiter en clair sur le réseau. Il est aussi possible de savoir à tout moment les pages Web consultées, les sessions ftp en cours, les mails envoyés où réceptionnés.

Cette technologie n'est pas forcement illégale car elle permet aussi de détecter des failles sur un système [6].

4. Les chevaux de Troie

Les chevaux de Troie sont des programmes informatiques cachés dans d'autres programmes.

En général, le but d'un cheval de Troie est de créer une porte dérobée (backdoor) pour qu'un pirate informatique puisse ensuite accéder facilement l'ordinateur ou le réseau informatique. Il peut ainsi voler des mots de passe, copier des données, exécuter des actions nuisibles. [6]

5. Man in the middle

L'attaque man in the middle (littéralement « *attaque de l'homme au milieu* » ou « *attaques de l'intercepteur* »), parfois notée *MITM*, est un scénario d'attaque dans lequel un pirate écoute une communication entre deux interlocuteurs et falsifie les échanges afin de se faire passer pour l'une des parties. La plupart des attaques de type « man in the middle » consistent à écouter le réseau à l'aide d'un outil appelé sniffer [5].

6. Les bombes logiques

Une Bombe logique est une partie d'un programme malveillant (virus, cheval de Troie, etc.) qui reste dormante dans le système hôte jusqu'à ce qu'un instant ou un événement survienne, ou encore que certaines conditions soient réunies, pour déclencher des effets dévastateurs en son sein [5].

7. Social enginnering

En utilisant les moyens usuels (téléphone, email…) et en usurpant une identité, un pirate cherche à obtenir des renseignements confidentiels auprès du personnel de l'entreprise en vue d'une intrusion future. Seule une formation du personnel permet de se protéger de cette attaque [5].

I.1.5. Politiques de sécurité :

Avant de mettre en place des mécanismes de protection, il faut préparer une politique à l'égard de la sécurité.qui fixe les principaux paramètres, c.-à-d. exprimer la volonté managériale de protéger les valeurs informationnelles et le ressources informatiques de l'organisation. Elle spécifie les moyens (ressources, procédures, outils, ...) et évite que le système d'information ne devienne une cible et cette protection peut être assuré par certains mécanismes de sécurité.

I.1.6. Mécanismes de sécurité des réseaux :

L'existence d'une grande variété d'outils d'attaque pousse le degré des risques d'intrusion à un taux plus élevé, ce qui incite les administrateurs à s'appuyer sur diverses solutions pour parer ces attaques. Parmi ces solutions, on peut citer :

a) Pare-feu

Un pare-feu, ou coupe-feu ou encore firewall, comme son nom l'indique, est un équipement dont l'objectif est de séparer le monde extérieur du monde

intérieur à protéger. Son rôle est de ne laisser entrer que les paquets dont l'entreprise est sûre qu'ils ne posent pas de problème. [7]

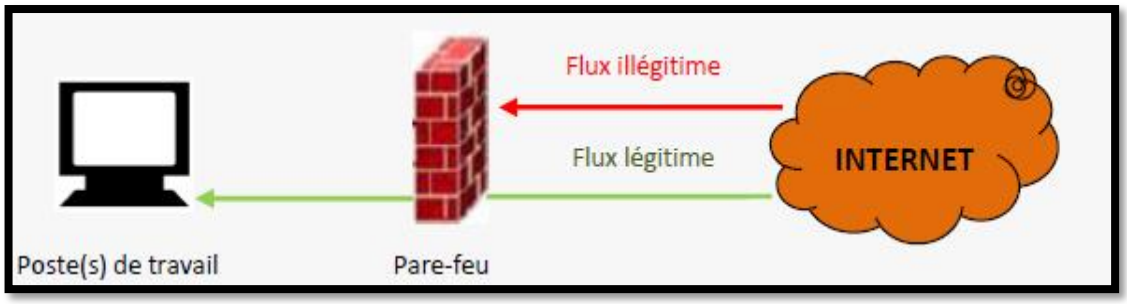

Figure I.8 : Control du flux entrant

L'envoi d'informations volées par un cheval de Troie est bloqué par le pare-feu.

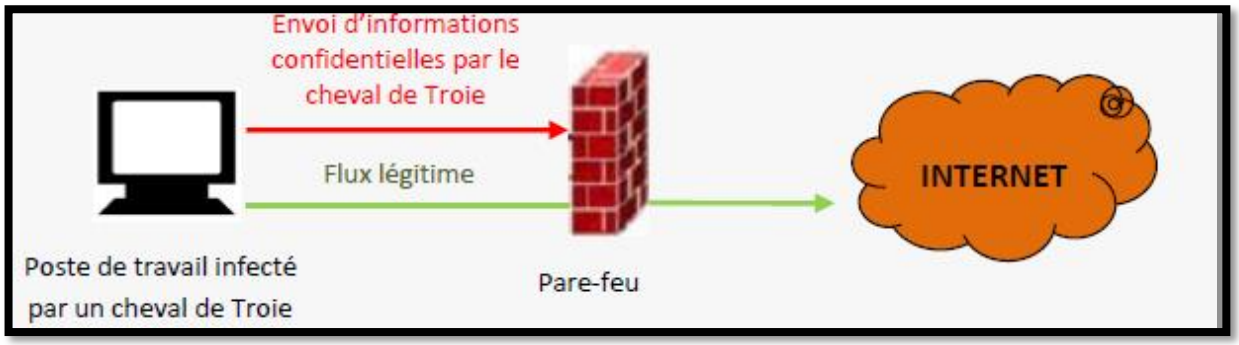

Figure I.9 : Control du flux sortant

b)Le NAT :

Le système de traduction d'adresses NAT (Network Address Translation) est apparu en 1994, initialement pour permettre la communication entre l'Internet et des réseaux privés contenants des adresses IP non conformes au plan d'adressage de l'Internet, et il a été ensuite très largement utilisé pour pallier le déficit d'adresses IP engendré par l'étroitesse de la plage d'adresses de IPv4. Il est devenu de ce fait à la fois une solution et un problème de sécurité des réseaux [8].

Le NAT consiste à établir des relations entre l'adressage privé dans un réseau et l'adressage public pour se connecter à Internet. Et d'autre part le mécanisme de translation d'adresse permet de sécuriser le réseau interne étant donné qu'il camoufle complètement l'adresse interne, en effet pour un observateur externe au réseau, toutes les requêtes semble provenir de la même adresse IP.

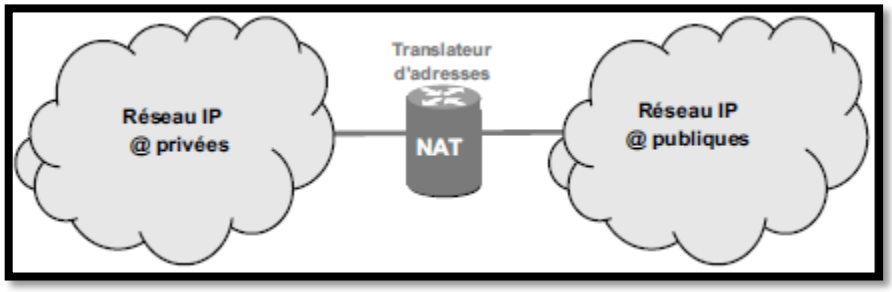

Figure I.10 : Position du NAT.

c) La DMZ :

Une DMZ (Demilitarized zone) est une zone tampon d'un réseau d'entreprise située entre le réseau local et Internet, derrière le pare-feu. Il s'agit d'un réseau intermédiaire regroupant des serveurs publics (HTTP, DHCP, mails, DNS, etc.).Ces serveurs doivent être accessibles depuis le réseau interne de l'entreprise et, pour certains, depuis les réseaux externes. Le but est ainsi d'éviter toute connexion directe au réseau interne. [2]

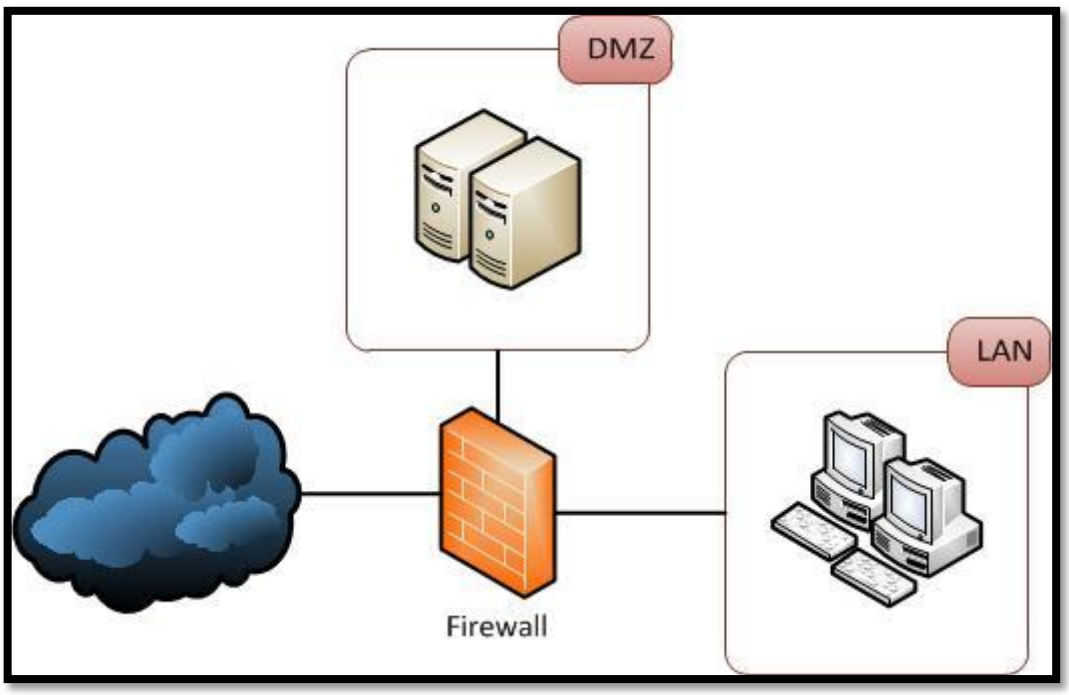

Figure I.11 : Emplacement d'une DMZ dans un réseau.

d)Les Proxys :

Un serveur Proxy (serveur mandataire) est à l'origine une machine ou logiciel faisant fonction d'intermédiaire entre le réseau local et Internet dans le but de rompre la connexion directe entre client est serveur, ce qui permette aux utilisateurs du réseau interne d'accéder à internet de manière plus sécurisé et sans que les utilisateurs externes ne soient capables d'accéder à ce réseau.

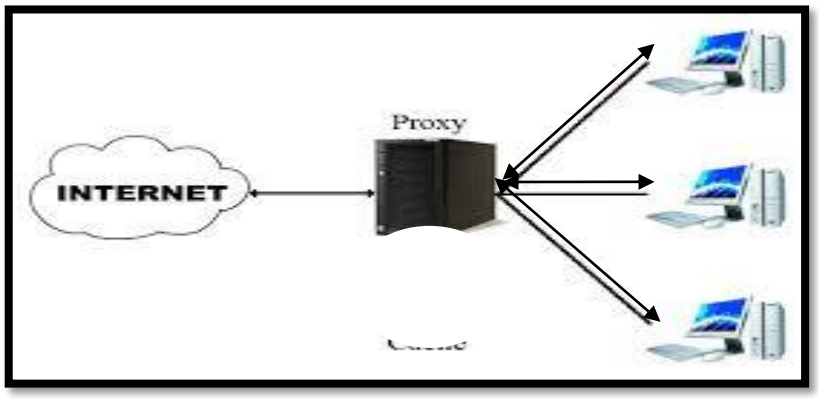

Figure I.12 : Principe d'un Proxy

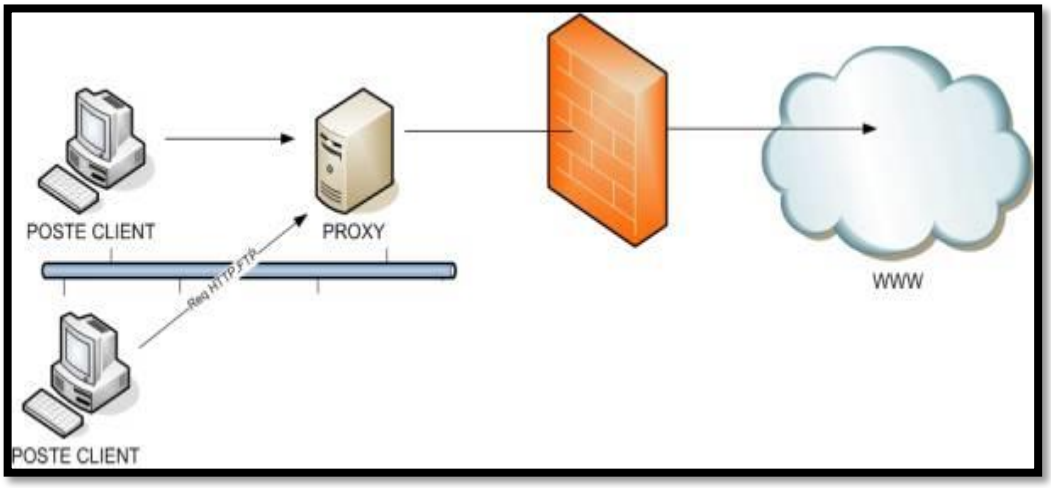

Figure I.13 : Emplacement d'un proxy dans un réseau

e) Les VPN (Virtual Privat Network) :

Un VPN est un tunnel, liaison virtuelle, sécurisé permettant la communication entre deux ou plusieurs LAN privés y compris au travers de réseaux peu sûrs comme Internet. Les données envoyées à travers de ces liaisons virtuelles sont chiffrées ; ce qui garantit aux utilisateurs d'un VPN qu'en cas d'interception malveillante les données soient illisibles.

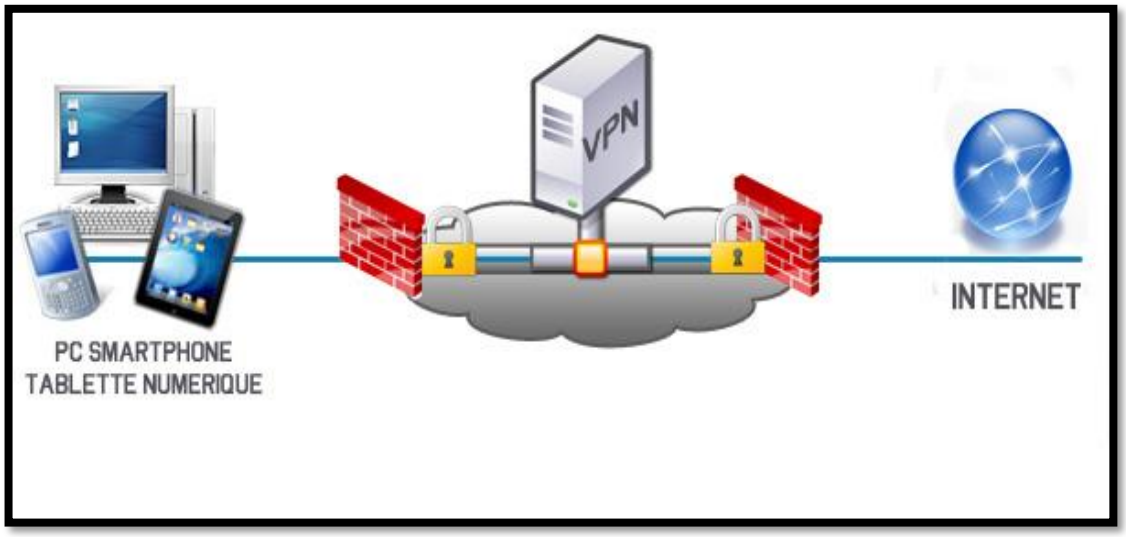

Figure I.14 : Principe d'un VPN

f) Les anti-virus :

Un antivirus est un programme capable de détecté la présence d'un processus malveillant sur un ordinateur et de le désinfecter si c'est possible. On parle ainsi d'éradication de virus pour désigner la procédure de nettoyage de l'ordinateur. [9]

I.2. Réseaux informatiques :

I.2.1. La norme OSI :

I.2.1.1. Définition

L'Open System Interconnection (OSI) est une norme établie par l'organisme Internationnal Standard Organisation(ISO), afin de permettre aux systèmes ouverts (ordinateur, terminal, réseau, etc.) d'échanger des informations avec d'autres équipements hétérogènes. Cette norme est constituée de 7 couches, dont les 4 premières sont dites basses et les 3 couches supérieures sont dites hautes. Le principe de fonctionnement est simple, chaque couche ne peut communiquer qu'avec la couche supérieure. Chacune des couches est composée d'éléments matériels et/ou logiciels chargés de « transporter » le message à la couche supérieure. [10]

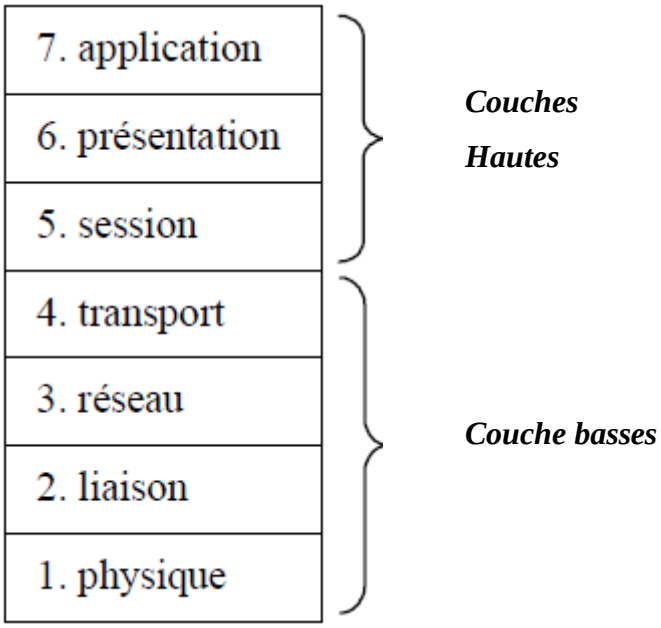

Figure I.15 : Le modèle OSI

I.2.1.2. Les couches du modèle OSI

a) La couche physique :

Définit les caractéristiques techniques, électrique, fonctionnelles et procédurales nécessaires à l'activation et à la désactivation des connexions physique destinées à la transmission de bits entres deux entité.

b) La couche liaison :

Définit les moyens fonctionnels et procéduraux nécessairesà l'activation et a l'établissement ainsi qu'au maintien et à la libération des connexions de liaisons de données entre les entités du réseau. Cette couche détecte et corrige, quand cela est possible, les erreurs de la couche physique et signale à la couche réseau les erreurs irrécupérables

c) La couche réseau :

Assure toutes les fonctionnalités de relais et d'amélioration de services entre les entités du réseau, c'est-à-dire : l'adressage, le routage, le contrôle de flux, la détection et la correction d'erreurs non résolues par la couche 2 (liaison) pour préparer le travail de la couche 4.

d) La couche transport :

Définit un transfert de données transparent entre les entités en les déchargeant des détails d'exécution (contrôle entre l'OS et le support de transmission). Son rôle est d'optimiser l'utilisation des services de réseau disponibles afin d'assurer à moindre coût les performances requise par la couche 5(session).

e) La couche session :

Fournit aux entités de la couche présentation les moyens d'organiser et de synchroniser les dialogues et les échanges de données. Il s'agit gestion d'accès, de sécurité et d'identification des services.

f) La couche présentation :

Assure la transparence du format des données à la couche 7 (application).

g) La couche application :

Assure aux processus d'application le moyen d'accès à l'environnement OSI et fournit tous les services directement utilisable par l'application (transfert de données, Allocation de ressources, intégrité et cohérence des informations, synchronisation des applications). [10]

I.2.2. Le model TCP/IP

I.2.2.1. Définition

TCP/IP désigne communément une architecture réseau, mais cet acronyme désigne en fait 2 protocoles étroitement liés : un protocole de transport, TCP (Transmission Control Protocol) qu'on utilise "par-dessus" un protocole réseau, IP (Internet Protocol). Ce qu'on entend par "modèle TCP/IP", c'est en fait une architecture réseau en 4 couches dans laquelle les protocoles TCP et IP jouent un rôle prédominant, car ils en constituent l'implémentation la plus courante. Par abus de langage, TCP/IP peut donc désigner deux choses : le modèle TCP/IP et la suite de deux protocoles TCP et IP.

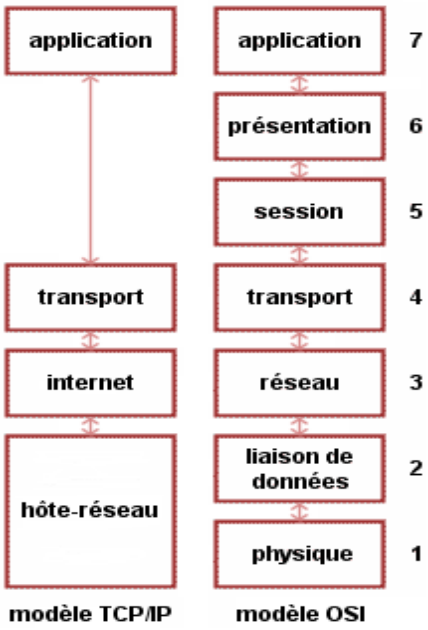

Figure I.16 : Les modèles TCP/IP et OSI

Conclusion

Nous avons présenté dans ce chapitre les différents aspects liés à la sécurité informatique, à savoir les différents types d'attaques et menaces, et et quelques mécanismes permettant de se protéger contre ces dernières.

Dans le chapitre qui suive, nous allons voir comment se protéger contre ce genre d'attaques en utilisant des logiciels spécialisés qui permettent de surveiller des données transitant sur un système et de réagir en cas de détection de données suspectes.

Introduction

De nos jours, l'ouverture des systèmes d'information à des échanges externes aux réseaux de l'entreprise donnent aux utilisateurs malveillants un moyen supplémentaire afin d'y attaquer ces dernier.

Afin de palier à ce problème, il est nécessaire d'avoir des logiciels spécialisés dont le rôle serait de surveiller les données qui transitent sur ce système, et qui serait capable de réagir si des données semblent suspectes. Les systèmes de prévention d'intrusions (IPS) et les systèmes de détection d'intrusions (IDS) conviennent parfaitement pour réaliser cette tâche.

Dans ce chapitre, nous présentons d'abord les concepts de base des IDS et des IPS, puis nous donnons leurs architectures et leurs principes de fonctionnement, enfin nous détaillerons leurs points forts et leurs points faibles.

II.1. Détection d'attaques :

La sécurité devient aujourd'hui un domaine très en vogue, il reste néanmoins très vaste et souvent complexe à maîtriser vu le nombre croissant d'attaque. Pour contrer ces attaques, plusieurs solutions de détection d'intrusions ont été mise en œuvre ; parmi ces solutions, on trouve les IDS et les IPS.

II.1.1. Système de détection d'intrusions (IDS)

Un IDS (Intrusion Detection Systems) est une sonde placée judicieusement sur un réseau ou un système afin de repérer les activités douteuses ou anormales sur cette cible et alerter les responsables sécurité. De cette façon, on peut obtenir une connaissance des tentatives réussies (ou non) d'attaque ou d'intrusion sur le système.

La détection d'intrusions est apparue au début des années 80, suite aux travaux d'Anderson 1975 et de Denning 1976, qui ont posé les fondations de la détection d'intrusions. Et ce n'est qu'au début des années 90 que les premiers produits commerciaux sont apparus.

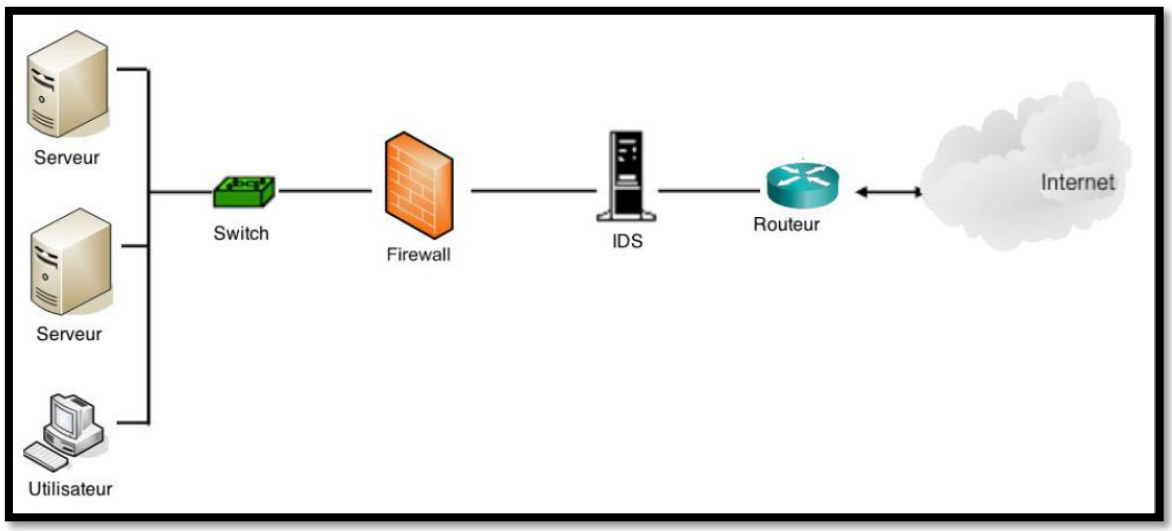

Figure II.1 : Emplacement d'un IDS dans un réseau.

II.1.2. Principe de fonctionnement d'un IDS

Les techniques de détection d'intrusion tentent de faire la différence entre une utilisation normale du système et une tentative d'intrusion et donnent une alerte. Typiquement, les données d'audit du système sont parcourues à la recherche de signatures connues d'intrusion, de comportements anormaux ou d'autres choses intéressantes. [13]

Un système IDS est placé en ligne et examine en théorie tous les paquets entrants ou sortants. Il réalise un ensemble d'analyses de détection, non seulement sur chaque paquet individuel, mais également sur les conversations et motifs du réseau, en visualisant chaque transaction dans le contexte de celles qui précèdent ou qui suivent. [4]

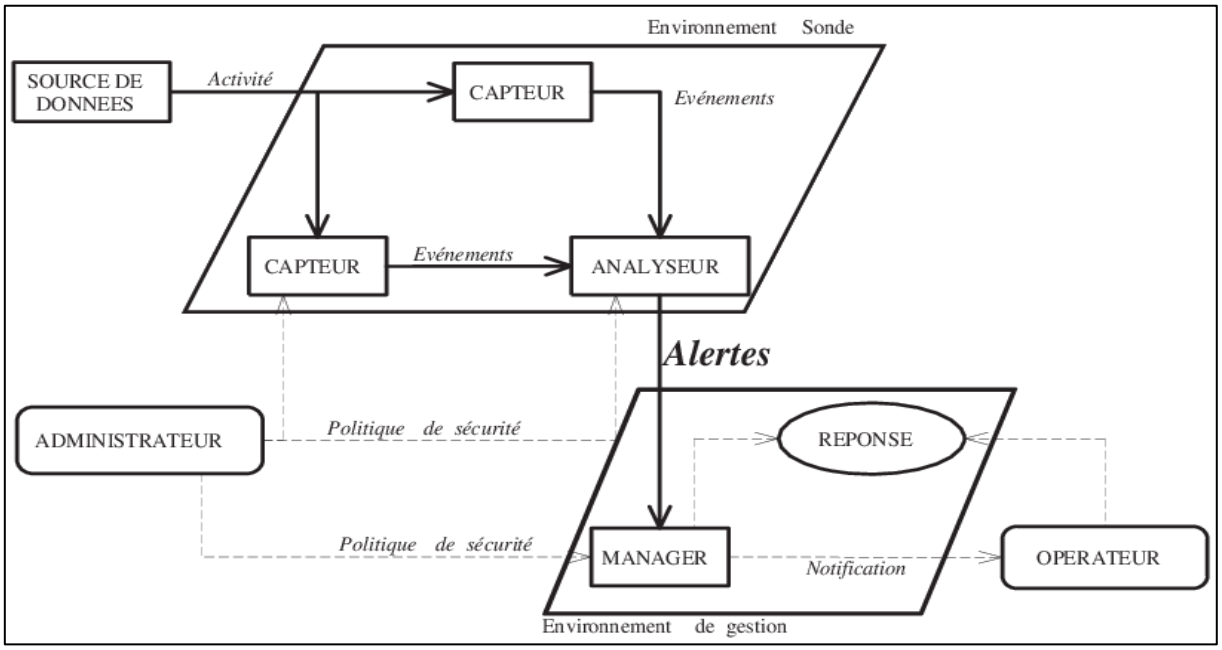

Figure II.2 : Schéma d'architecture IDWG d'un IDS.

Comme le montre la figure ci- dessous, un IDS est composé de plusieurs éléments dont chacun accompli un rôle bien précis.

- **Capteur**

Le capteur observe l'activité du système et génère des évènements qui renseignent de l'évolution de l'état du système en filtrant des données provenant d'une source de données.

- **Analyseur**

L'analyseur met en œuvre l'approche choisie pour la détection. Il génère des alertes lorsque le flux d'évènements fourni par le capteur contient des éléments caractéristiques d'une activité malveillante.

- **Manager**

Le manager collecte les alertes produites par l'analyseur, et les transmit ensuite à l'opérateur sous forme de notifications afin de lui permettre de gérer les alertes reçues et prendre des décisions.

- **Opérateur**

Personne chargée de l'utilisation de manager associé à l'IDS. Elle propose ou décide de la réaction à prendre en cas d'alerte : c'est parfois la même personne que l'administrateur.

- **Administrateur**

Personne chargée de mettre en place la politique de sécurité, et par conséquent de déployer et de configurer les IDS.

II.1.3. Critères de classification des IDS

Avec leur utilisation dans des environnements et des contextes très variés, plusieurs critères de classification des IDS ont été proposés.

Approche de détection

Les IDS peuvent être répertoriés dans deus grandes approches de détection. Une approche basée sur les scénarios, elle utilise des bases de signatures d'attaque pour la recherche d'intrusion. La deuxième approche est basée sur le comportement du système vis-à-vis des intrusions. Notons que cet aspect est lié au module d'analyse de l'IDS [12].

Source de données

Les données utilisées dans l'IDS jouent un rôle très important dans le mécanisme de détection d'intrusions. C'est l'interface entre l'IDS et le système surveillé. Les données traitées peuvent provenir d'un trafic réseau ou bien des fichiers locaux du système d'exploitation, comme ça peut provenir des fichiers traités par uns application. Il y a aussi des IDS qui utilisent plusieurs sources de données, c'est une forme hybride. Dans l'architecture d'un IDS, cet aspect est directement lié au module de capture ou bien à la source de données [12].

Comportement après détection

Si l'IDS détecte une attaque, deux comportement peuvent êtres adoptés, une réponse passive ou active. Cet aspect est souvent lié au module de réponse de l'IDS [12].

Fréquence d'utilisation

Cet aspect dépend du mode du fonctionnement du module d'analyse de l'IDS. Deux modes d'analyse peuvent êtres assurés par un IDS : continu ou bien par lots [12].

Stratégie de contrôle

Elle décrit les méthodes de gestion des modules composant l'IDS, ainsi que les modes de contrôle appliqués aux entrées et aux sorties de l'IDS. Trois stratégies peuvent êtres appliquées aux IDS : centralisée, partiellement distribuée ou bien entièrement distribuée [12].

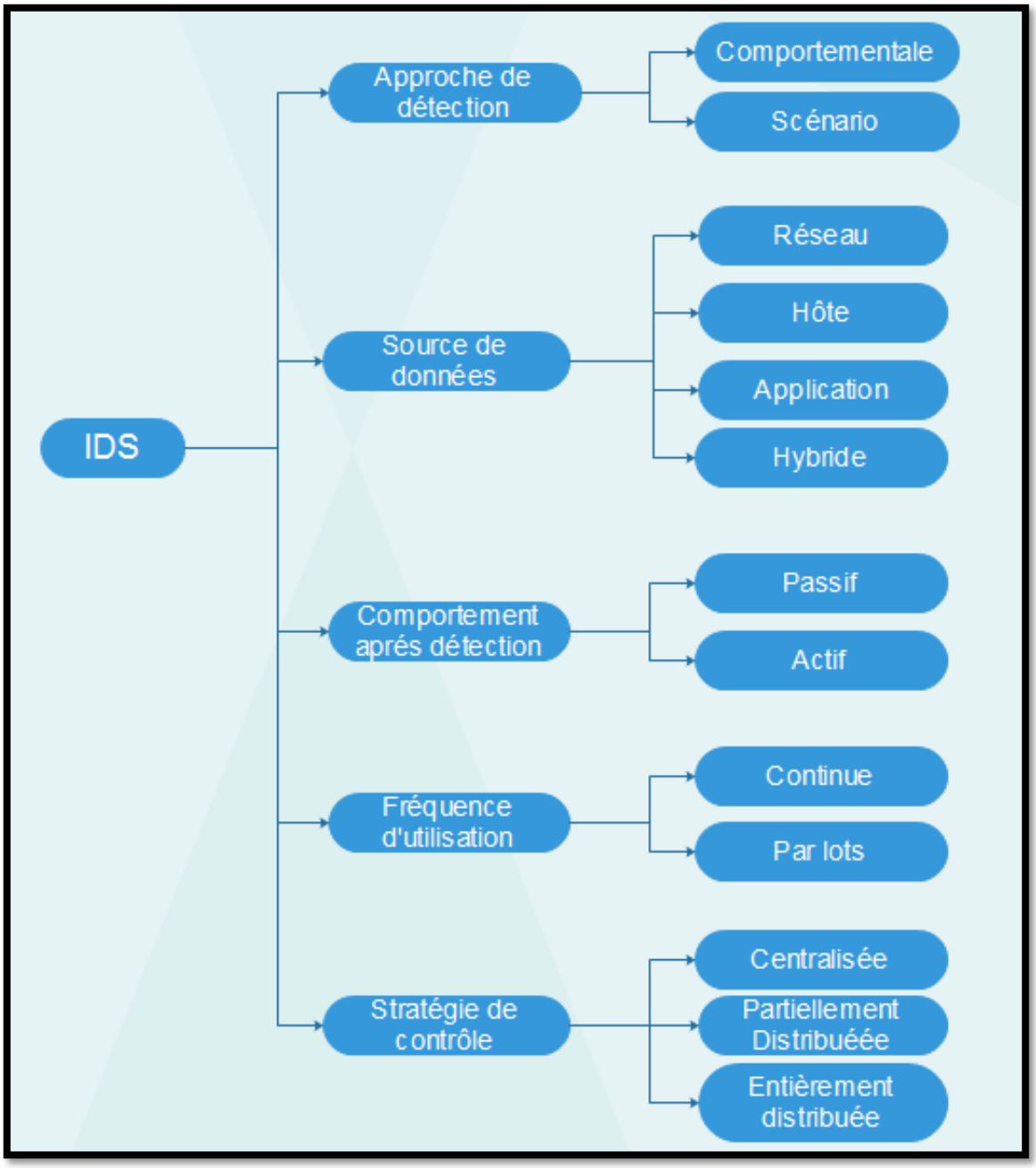

Figure II.3 : Classification des systèmes de détection d'intrusion.

II.1.4. Les types d'IDS

Comme nous l'avent vus dans le chapitre précédent, les attaques que peux subir le system varie selon la faille (Réseau ou Programmation), de ce fait la détection d'intrusion peut se faire à plusieurs niveaux. Les IDS peuvent être classés en trois catégories majeures selon l'activité qu'ils s'attachent à surveiller :

a) Les NIDS (Network Intrusion Detection System)

Les IDS réseaux (Network IDS) analysent en temps réel le trafic qu'ils aspirent à l'aide d'une sonde (carte réseau en mode "promiscuous"). Ensuite, les paquets sont décortiqués et analysés.

Il est fréquent de trouver plusieurs IDS sur les différentes parties d'un réseau. On trouve souvent une architecture composée d'une sonde placée à l'extérieur du réseau afin d'étudier les tentatives d'attaques et d'une sonde en interne pour analyser les requêtes ayant traversé le pare-feu. [13]

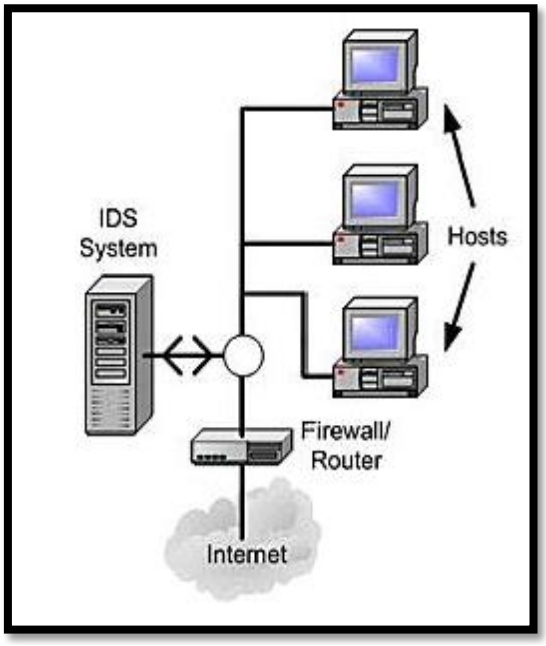

Figure II.4 : Déploiement du NIDS.

Avantages des NIDS : Les NIDS ont plusieurs avantages, les plus importants sont :

- Un NIDS peut surveiller un grand réseau ;
- Le déploiement de NIDS a peu d'impact sur un réseau existant. Ils sont habituellement des dispositifs passifs, qui écoutent sur un fil de réseau sans interférer l'opération normale du réseau. Ainsi, il est habituellement facile de monter en rattrapage un réseau pour inclure IDS avec l'effort minimal.
- Un NIDS peut être très sûr contre les attaques et être même caché pour beaucoup d'attaquants.

Les inconvénients des NIDS : Malgré leurs avantages, les NIDS présentent certains inconvénients à savoir :

- Un NIDS ne peut pas traiter tous les paquets circulant sur un grand réseau, et il ne peut pas reconnaître des attaques pendant le temps de haut trafic.
- Quelques fournisseurs essayent à implémenter l'IDS dur le matériel pour qu'il marche plus rapidement.
- Plusieurs avantages des NIDS ne peuvent pas être obtenus sur les commutateurs modernes. La plupart des commutateurs ne fournissent pas des

surveillances universelles des ports et limitent la gamme de surveillance de NIDS. Même lorsque les commutateurs fournissent de tels ports de surveillance, souvent les ports simples ne peuvent pas refléter tout le trafic traversant le commutateur.

- Un NIDS ne peut pas analyser les informations chiffrées (cryptées) dans les organisations utilisant un VPN.
- La plupart des NIDS ne peuvent pas indiquer si une attaque est échoué ou non. Ils reconnaissent seulement qu'une attaque est initialisée. C'est-à-dire qu'après que le NIDS détecte une attaque, l'administrateur doit examiner manuellement chaque host s'il a été en effet pénétré [14].

b) Les HIDS (Host Intrusion Detection System):

Les IDS systèmes (Host IDS) analysent le fonctionnement et l'état des machines sur lesquelles ils sont installés afin de détecter les attaques en se basant sur des démons (tels que syslogd par exemple). L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levées [13].

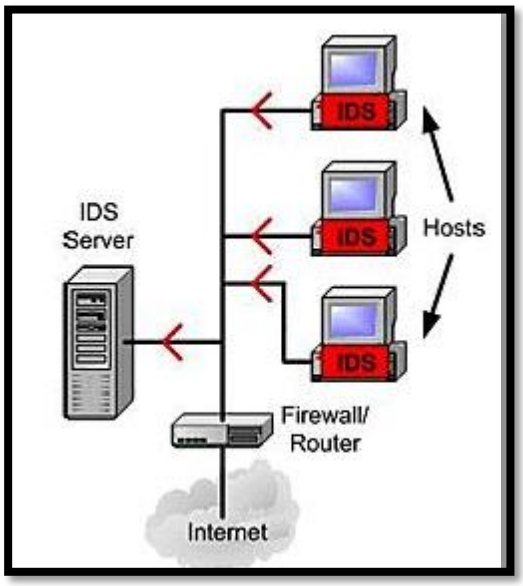

Figure II.5: Déploiement du HIDS

Avantages des HIDS : Parmi les avantages des HIDS, on cite :

- Surveillance des événements locaux jusqu'au host, et détection des attaques qui ne sont pas vues par le NIDS ;
- Marcher dans un environnement dans lequel le trafic de réseau est encrypté, lorsque les sources des informations de host-based sont générées avant l'encrypte des données ou après le décrypte des données au host de la destination ;
- Un HIDS n'est pas atteint par un réseau commuté ;
- Lorsque les HIDS marchent sur la traîné de l'audit de SE, ils peuvent détecter le Cheval de Troie ou les autres attaques relatives à la brèche intégrité de logiciel.

Inconvénients des HIDS: De même que les NIDS, les HIDS présentent quelques inconvénients qui sont :

- Un HIDS est difficile à gérer : des informations doivent être configurées et gérées pour chaque host surveillé ;
- Puisque les sources d''information de HIDS résident sur l'host qui est la cible de la attaques, l'IDS peut être attaqué et neutralisé comme une partie de l'attaque.
- Un HIDS n'est pas bon pour la surveillance du réseau entier parce qu'il ne voit que les paquets du réseau reçus par ses hosts.
- Un HIDS peut être neutralisé par certaine attaque de typeDoS.
- Lorsque HIDS emploie la traîné de l'audit du SE comme des sources d'informations [14].

c) Les IDS Hybrides (NIDS + HIDS)

Un "hybride" IDS est une sorte de tout en un, c'est un HIDS avec un NIDS. Ce nom peut aussi s'appliquer à une solution mêlant plusieurs IDS, ou des IDS particuliers

Les IDS hybrides rassemblent les caractéristiques de plusieurs IDS différents. En pratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil, de surveiller les réseaux et les terminaux. Les sondes sont placées en des points stratégiques et agissent comme des NIDS et/ou des HIDS suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout, et agréger les informations d'origines multiples.

Les avantages des IDS hybrides sont multiples :

• Moins de faux positifs ;

• Meilleure corrélation (la corrélation permet de générer de nouvelles alertes à partir de celles existantes) ;

• Possibilité de réaction sur les analyseurs [13].

II.1.5. Méthode de détection d'intrusion

La détection d'intrusions repose sur deux approches de base : L'approche comportementale et l'approche basée connaissance.

a) L'analyse comportementale

Dans l'analyse comportementale, un modèle de comportement normal du système surveillé est préalablement construit. Ce modèle est appelé profil de comportement normal qui sera utilisé comme une référence dans la détection. Au cours de la surveillance du système, toute déviation significative du comportement courant de système contrôlé par rapport au comportement normal de référence donne lieu à une attaque. Cette approche possède un certain nombre d'avantages et d'inconvénients [11].

- **Avantages de l'analyse comportementale :**
  - Elle n'exige pas des connaissances préalables sur les attaques ;
  - Elle permet la détection de la mauvaise utilisation des privilèges ;

- Elle permet de produire des informations qui peuvent être employées pour définir des signatures pour l'analyse basée connaissance [11].

➢ **Les inconvénients de l'analyse comportementale :**

- Elle produit un taux élevé des alarmes de type positif faux en raison des comportements imprévisibles des utilisateurs du réseau.
- Cette approche nécessite des phases d'apprentissage pour caractériser les profils de comportement normaux ;
- Les alarmes générées par cette approche ne sont pas significatives.

Cette étude comparative entre les deux approches d'analyses utilisées par les systèmes de détection d'intrusions montre l'existence d'une complétude entre ces deux méthodes. Cette complétude qui permettra de surmonter les inconvénients relatifs à chaque méthode d'analyse. Pour cette raison, il est préférable d'adopter les deux techniques d'une manière parallèle pour obtenir un système de détection d'intrusions efficace. Cependant, les systèmes de détection d'intrusions commerciaux disponibles emploient seulement la technique basée signature, ce qui motive les efforts de recherche croissants pour construire des détecteurs d'anomalies efficaces pour des buts de détection d'intrusions. L'effort principal de cette recherche est concentré sur les systèmes de détection d'intrusions qui sont basés sur la technique comportementale. Pour cette raison, nous présenterons dans la section suivante les différentes approches utilisées dans la méthode de détection comportementale [11].

b) L'approche basée connaissance

Cette approche de détection est désignée en anglais par le terme « MisuseDetection », qui signifie dans la littérature la détection d'une mauvaise utilisation, et il existe plusieurs traductions françaises adoptées pour cette approche, par exemple l'approche par signatures ou par scénarios.

Elle est caractérisée par l'existence d'une base de connaissances qui comporte des modèles d'attaque connus a priori qui sont appelés les signatures. Elle examine les activités du système et du réseau en cherchant des événements ou l'ensemble des événements qui décrivent une attaque connue. Ainsi, dans cette approche, tous ce qui n'est pas explicitement interdit est autorisé. Cette approche possède aussi un certain nombre d'avantages et d'inconvénients [11].

➢ **Avantages de l'analyse basée connaissance :**

- L'analyse basée connaissance est très efficace pour la détection d'attaques avec un taux très bas des alarmes de type positif faux ;
- Les alarmes générées sont significatives [11].

➢ **Inconvénients de l'analyse basée connaissance**

L'analyse basée connaissance permet seulement la détection des attaques qui sont connues au préalable. Donc, la base de connaissances doit être constamment mise à jour avec les signatures de nouvelles attaques.

Le risque que l'attaquant peut influencer sur la détection après la reconnaissance des signatures [11].

I.1.6. Méthodes d'analyse

Analyse centralisée

Certains IDS ont une architecture multi-capteurs (ou multisondes). Ils centralisent les événements (ou alertes) pour analyse au sein d'une seule machine. L'intérêt principal de cette architecture est de faciliter la corrélation entre événements puisqu'on dispose alors d'une vision globale. Par contre, la charge des calculs (effectués sur le système central) ainsi que la charge réseau (due à la collecte des événements ou des alertes) peuvent être lourdes et risquent de constituer un goulet d'étranglement. [15]

Analyse locale

Si l'analyse du flot d'événements est effectuée au plus près de la source de données (généralement en local sur chaque machine disposant d'un capteur), on minimise le trafic réseau et chaque analyseur séparé dispose de la même puissance de calcul. En contrepartie, il est impossible de croiser des événements qui sont traités séparément et l'on risque de passer à côté de certaines attaques distribuées. [15]

Analyse distribuée

Partiellement distribuée : Dans ce cas un nombre limité de nœuds peuvent exécuter des taches d'analyse locale et de détection mais ils sont commandés par un nœud maitre, celui-ci collabore avec d'autres nœuds maitres pour superviser la détection globale sous forme d'une structure hiérarchique. [12]

Entièrement distribuée : La collecte d'informations, l'analyse et la détection ainsi que les alertes seront réalisées au niveau local de chaque nœud. Mais dans le cas d'information incomplète ou bien suspicion les nœuds peuvent déclencher des procédures de collaboration supervisées par des nœuds maitres. [12]

Analyse hiérarchique

Cette méthode est utilisée pour résoudre les problèmes de scalabilité (mise à l'échelle) lies aux méthodes centralisées. Cette architecture utilise une structure hiérarchique sous forme d'arbre. Les données collectées localement traversent hiérarchiquement les nœuds de l'arbre. A chaque niveau, un traitement peut se faire sur les

données pour détecter un type d'intrusion ou bien pour réduire la quantité d'informations transmises au niveau supérieur. Cette approche est tolérante aux pannes, donc si un nœud supérieur tombe en panne les nœuds de niveau inferieur peuvent assurer une détection dans des conditions minimales. [12]

I.1.7. Les fonctions proposées par les IDS:

- Détection d'attaques (actives ou passives).
- Génération des rapports.
- Outils de corrélation avec d'autres éléments de l'architecture de sécurité.
- Réaction aux attaques par le blocage de route ou la fermeture de connexion.
- Transfert d'activités.

I.1.8. Efficacité des IDS

L'efficacité d'un système de détection d'intrusions est déterminée par les mesures suivantes:

- **Exactitude** : Le système de détection d'intrusions n'est pas exact s'il considère les actions légitimes des utilisateurs comme atypiques ou intrusives.
- **Performance** : La performance d'un système de détection d'intrusions est mesurée par le taux de traitement des traces d'audits. Si la performance du système de détection d'intrusions est pauvre, donc la détection en temps réel n'est pas possible.
- **Perfection** : Un système de détection d'intrusions est imparfait s'il n'arrive pas à détecter une attaque.
- **Tolérance aux pannes** : Un système de détection d'intrusions doit être résistant aux attaques, en particulier dans le cas des attaques de déni de service.

II.2. Système de prévention d'intrusion(IPS)

IPS est un autre concept qui a fait son apparition au début des années 2000 sous l'idée qu'un système de détection d'intrusion, qui peut détecter des attaques contre un réseau mais sans pouvoir l'empêcher. Cela a mené certaines entreprises utilisatrices à se poser la question : pourquoi s'investir dans une solution de détection d'intrusions si on ne peut pas empêcher l'intrusion ? La réaction des fournisseurs a été rapide ; et c'est ainsi que le concept IPS a vu le jour.

Un système de prévention d'intrusion est un ensemble de composants logiciels et/ou matériels dont la fonction principale est d'empêcher toute activité suspecte détectée au sein d'un système [16].

II.2.1 Les systèmes de prévention d'intrusions réseau NIPS

a) Définition

Lors de la détection d'une attaque, le système réagit et modifie l'environnement du système attaqué. Cette modification peut être le blocage de certains flux, de certains ports ou l'isolation pure et simple de certains systèmes du réseau. Le point sensible de ce genre de dispositif de prévention est qu'en cas de faux positif alors c'est le trafic du système qui est directement affecté. Les erreurs doivent donc être les moins nombreuses possibles car elles ont un impact direct sur la disponibilité des systèmes. En cas de détection de trafic dangereux lié à une intrusion potentielle, l'IPS bloque ce trafic comme un firewall [14].

b) Fonctionnement d'un NIPS

Le NIPS combine les caractéristiques d'un IDS standard avec celles d'un firewall. On le qualifie parfois de firewall à inspection en profondeur (deep inspection).

Comme avec un firewall, le NIPS a au minimum deux interfaces réseau, une interne et une externe. Les paquets arrivent par une des interfaces et sont passés au moteur de détection. L'IPS fonctionne comme un IDS en déterminant si paquet est malveillant ou non. Cependant, en plus de déclencher une alerte dans le cas où il détecte un paquet suspect, il rejettera le paquet et marquera cette session suspecte. Ainsi, quand des paquets arriveront après cette session à l'IPS ils seront rejetés.

Les NIPS sont déployés en ligne avec le segment du réseau à protéger, du cout toutes les données qui circulent entre le segment surveillé et le reste du réseau sont forcés de passer par le NIPS. Un NIPS déclenche des alarmes du type ' tel ou tel trafic a été détecter en train d'essayer d'attaquer ce système et a été bloqué [17].

c) Avantage d'un NIPS

- **Blocage rapide des intrusions :** En intervenant dès la détection, un système IPS bloque rapidement l'intrusion et minimise la durée totale avant que le réseau ne revienne à la normale ;
- **Détection précise et fiable :** A l'aide de plusieurs méthodes de détection et tirant parti de sa position en ligne, le système IPS peut détecter les attaques et intrusions avec une précision et une fiabilité supérieures ;
- **Prévention active :** Alors qu'un système NIDS prévient simplement de la présence d'un trafic suspect ou anormal, un système IPS peut lancer divers mécanismes de réaction [17].

II.2.2 Les systèmes de prévention d'intrusion du Hote (HIPS)

Un système de prévention d'intrusions sur l'hôte ou HIPS (Host Intrusion Prevention System) est destiné à arrêter les malwares, avant qu'une mise à jour de la détection spécifique ne soit publiée, en surveillant le comportement du code. La majorité des solutions HIPS surveillent le code lors de son exécution et interviennent si le code est considéré suspect ou malveillant [14].

II.2.3 Architecture d'un IPS

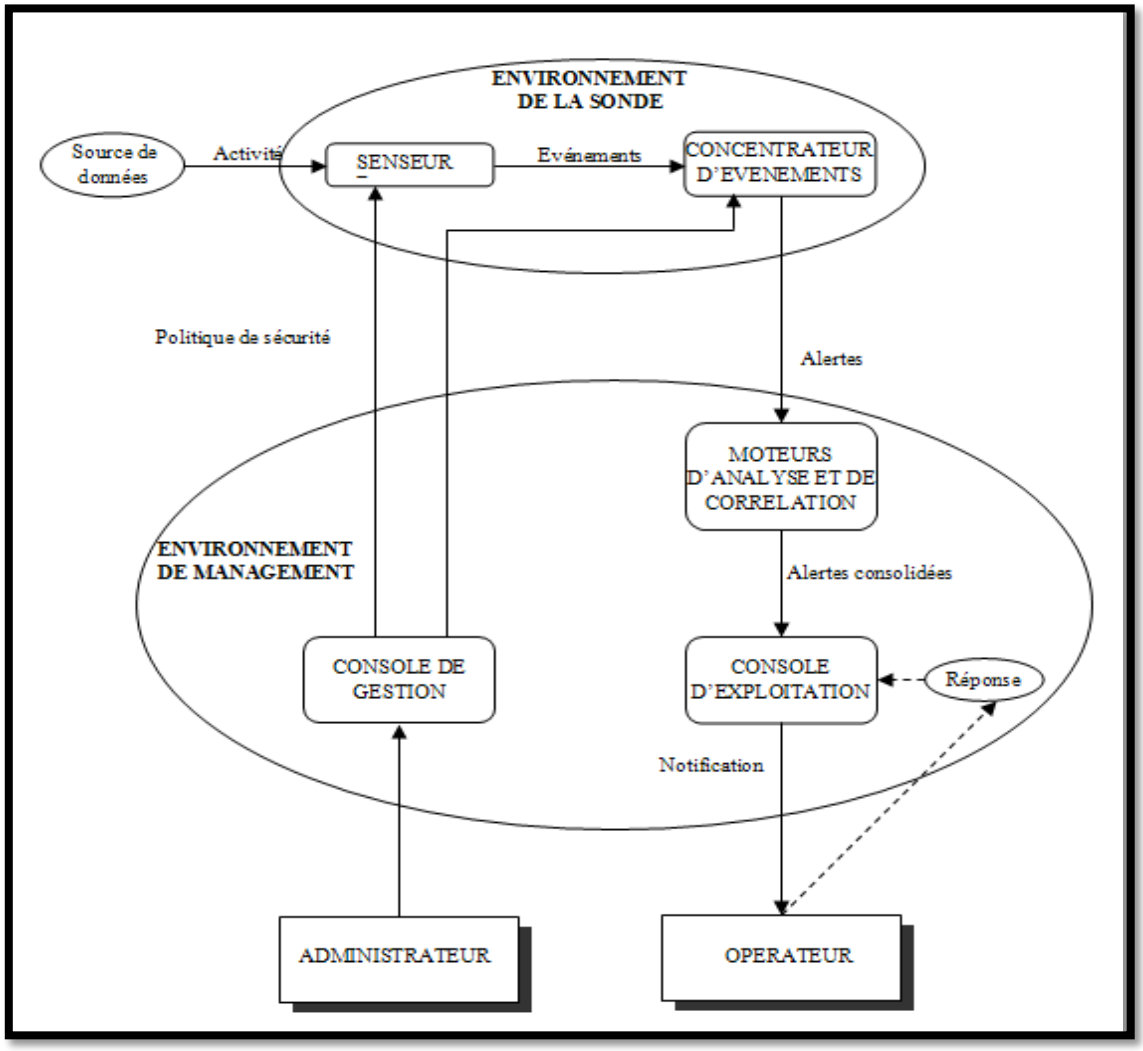

Figure II.6 : Architecture d'un IPS

 Activité : les flux d'activité émis par la source sont des éléments de données bruts qui sont identifiés par le senseur. Ils peuvent être intéressants pour l'opérateur.

 Evénement : il s'agit d'une activité détectée par le senseur et qui peut éventuellement être convertie en alerte IDMEF (*Intrusion Detection Message Exchange Format*) avant d'être transmise.

 Alerte : il s'agit d'un message émis d'un analyseur vers un manager et qui indique qu'un événement intéressant a été détecté. Une alerte contient généralement des informations concernant l'activité détectée ainsi que des informations complémentaires concernant les occurrences de cette activité.

 Notification : c'est le procédé qui permet au manager d'alerter l'opérateur d'une alerte.

 Administrateur: il s'agit de la composante humaine qui a la charge d'adapter les règles des IPS de manière à ce qu'elles se conforment à la politique de sécurité de l'entreprise.

 Opérateur : il s'agit de la personne qui est en charge de l'exploitation des IPS et notamment de la création des rapports et de l'application éventuelle des mécanismes de réponse à des alertes et des notifications.

 Senseur: il s'agit de l'entité qui collecte les flux de données, détecte des événements et les passe à l'analyseur.

 Analyseur: c'est l'entité qui analyse les événements et qui génère des alertes conformément à la politique de sécurité appliquée par l'administrateur.

 Manager : c'est le composant à partir duquel l'opérateur gère et interroge les différents composants du système du globale. Le manager utilise des notifications pour informer l'opérateur que des alertes ont eu lieu [14].

II.3. Comparaison entre IDS et IPS

HIDS	HIPS
Avantages : - Alerte sur des changements au niveau du système. - Détection de l'utilisation d'un système violant la politique de sécurité de l'entreprise. **Inconvénients:** - Un cout élevé de déploiement et de la gestion - Une détection réussie vient d'une tentative réussie d'attaques.	**Avantages :** - Assurer la protection contre les attaques inconnues. - Exige peut ou aucune mise à jour dans une période annuelle. - Détecter et empêcher les attaques de s'exécuter sur une machine au niveau noyau. **Inconvénients:** - Le temps de déploiement peut être long afin d'équiper chaque serveur et/ou post de travail. - Le produit nécessite un ajustement après l'installation pour être outil de sécurité.

Tableau II.1 : Comparaison entre les HIDS et les HIPS

NIDS	NIPS
Avantages : - Capable de détecter des anomalies même sur les systèmes qui emploient le cryptage. - L'observation du trafic avec un système basé sur des règles peut aider à imposer une utilisation du réseau en respectant la politique de l'entreprise. **Inconvénients :** - A moins qu'un plan de réponse ne soit conçu et mis en place, l'IDS fournit peu ou aucune sécurité. - Un déploiement réussi demande un important ajustement de l'IDS pour réduire au minimum les faux positifs.	**Avantages :** - Peut arrêter la propagation des vers, s'il est déployé correctement, sans arrêter le trafic. - Protège contre les nouvelles attaques avant que le code d'exploit soit sorti. -réduire le cout de la réponse aux incidents. **Inconvénients :** - Le coût du déploiement d'un NIPS au sein d'un réseau peut être important. - Un NIPS nécessite toujours des mises à jour de sécurité pour être vraiment efficace.

Tableau II.2 : Comparaison entre les NIDS et les NIPS

Conclusion

Dans ce chapitre, nous avons présenté les systèmes de détection d'intrusion à savoir leurs fonctionnements et leurs architectures. Il nous est paru évident que ces systèmes sont à présent indispensables aux entreprises afin d'assurer leur sécurité informatique en complétant les tâches des autres équipements de sécurité. Dans le chapitre suivant, nous allons voir comment les mettre en œuvre afin de mieux sécuriser un réseau.

Introduction

Plusieurs IDS ont été proposées dans la littérature : Haystack, MIDAS, IDES, NIDES, USTAT, IDIOT, GrIDS, GASSATA, Snort, etc. Ces IDS utilisent des techniques différentes dans le but d'améliorer leurs capacités de détection d'intrusions.

Dans ce chapitre, nous présentons quelques-uns de ces IDS d'une manière générale sans donner beaucoup de détails, en se contentant de donner pour chacun d'eux son principe de fonctionnement de base. Pour avoir plus de détail sur chaque algorithme, on peut se référer à la source d'où il est tiré.

III.1. Haystack

Le prototype Haystack [18] a été conçu pour détecter des intrusions sur un système multi-utilisateurs. Il réduit la traîné de l'd'audit du système à des comportements des utilisateurs, à des événements anormaux et à des incidents de sécurité. Il était réalisé pour détecter six types d'intrusions [1] :

1. Quand un utilisateur non-autorisé tente d'accéder au système ;
2. Quand un utilisateur autorisé tente de prendre l'identité d'un autre ;
3. Quand un utilisateur tente de modifier les paramètres relatifs à la sécurité du système ;
4. Quand un utilisateur tente d'extraire des données potentiellement sensibles du système ;
5. Quand un utilisateur bloque l'accès aux ressources du système à d'autres utilisateurs ;
6. diverses attaques telles que l'effacement de fichier, etc.

Pour parvenir à ses fins, Haystack utilise deux méthodes de détection : par détection d'anomalies et par signatures. La détection d'anomalies utilise un modèle par utilisateur décrivant le comportement de cet utilisateur dans le passé et un stéréotype qui spécifie le comportement générique acceptable pour cet utilisateur, évitant une dérive trop importante du premier modèle utilisateur. Il est ainsi impossible à un intrus d'habituer le système à un comportement intrusif.

III.1.1. Structure conceptuelle de *Haystack* [18]

Le système ***Haystack*** est composé de deux programmes, l'un s'exécutant sur le mainframe Unisys (Sperry) 1100/60 et l'autre sur le PC 2-248. Au plus haut niveau, Haystack interagit avec trois entités externes : le système d'exploitation Unisys (Sperry) 1100, l'agent de sécurité

du système et le système de gestion de la base de données sur la plate-forme d'analyse. La structure conceptuelle du système Haystack est représentée dans la figure suivante :

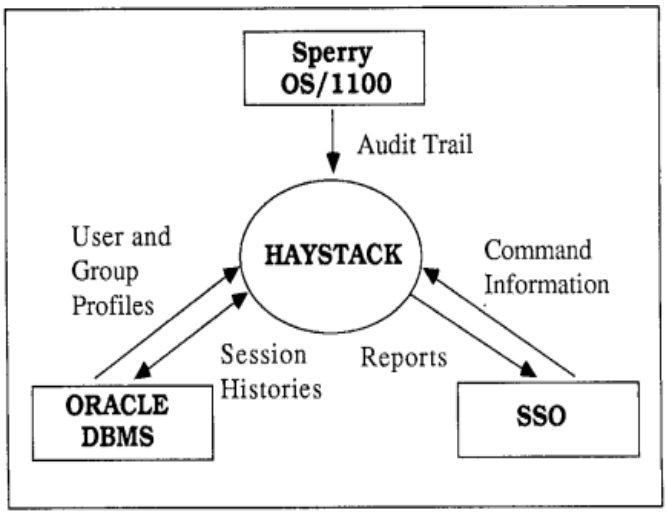

Figure 3.1. Structure conceptuelle de *Haystack*

a) Unisys (Sperry) :

Les programmes Unisys (Sperry) extraient la traîné de l'audit du journal du système d'exploitation, les analysent en fonction des éléments abstraits qui constituent l'événement de la traîné de l'audit généralisée, les transforment en fichiers, et les écrit dans un format standart ANSI.

b) L'agent de sécurité Système (SSO)

Le SSO peut sélectionner des utilisateurs particuliers et des intervalles de temps pour l'analyse, ou traiter l'ensemble de l'information de la traîné de l'audit.

Les programmes PC Haystack constituent la plupart du code du système. Ils implémentent quatre fonctions principales sur le PC.

III.2. MIDAS (Multics Intrusion Detection and Alerting System) [1]

MIDAS [19] est basé sur le concept de détection d'intrusion heuristique, qui consiste à définir des valeurs seuil pour certaines caractéristiques. Les auteurs s'inspirent d'un raisonnement d'un administrateur humain en analysant comment il mènerait une analyse sur des journaux d'audit pour trouver des preuves d'intrusion. Un administrateur humain peut faire plusieurs hypothèses sur le déroulement des intrusions ce qui lui va permettre de réduire considérablement les évènements à analyser.

MIDAS est un système expert à base des règles, qui applique un certain raisonnement. Il utilise le Production Based Expert System Toolset et des règles selon ces trois catégories suivantes :

a) Attaque immédiate : Les heuristiques d'attaque immédiate sont statiques, elles ne changent qu'avec l'intervention de la personne chargée de la sécurité. Elles opèrent sans aucune connaissance de l'historique du système, sur une très petite fenêtre d'événements.

b) Anomalie d'un utilisateur : Les classes de règles pour les anomalies d'utilisateurs utilisent les profils statistiques des comportements passés des utilisateurs. Deux profils sont maintenus: le profil de session qui n'est valable que pendant la session courante et le profil utilisateur qui dure sur une longue période de temps. Le profil de session est mis à jour au login de l'utilisateur à partir de son profil utilisateur, qui lui est mis à jour à son tour à partir du profil de session au logout.

c) État du système : Les heuristiques de l'état du système maintiennent des informations sur les statistiques du système en général, sans intérêt particulier pour les utilisateurs individuels, comme par exemple le nombre total de login ratés par opposition au nombre de logins réussit d'un utilisateur particulier. Les auteurs ont montré que MIDAS est assez rapide pour l'analyse en temps réel, mais génère trop de fausses alarmes.

III.3. IDES (Intrusion Detection Expert System)

IDES [20] se base sur l'hypothèse que le comportement des utilisateurs est presque constant dans le temps, et que la manière dont ils se comportent peut être résumée à certains comportements. IDES construit ses profils par groupes d'utilisateurs ayant un comportement proche et tente de corréler le comportement actuel d'un utilisateur avec son comportement passé et le comportement passé du groupe. Il observe trois types de sujets : les utilisateurs, les hôtes distants et les systèmes cibles. Au total, 36 paramètres sont mesurés : 25 pour les utilisateurs, 6 pour les hôtes et 5 pour les systèmes cible. Toutes ses mesures font partie de ces deux catégories **[**1**]** :

A) Mesure catégorique : C'est une mesure de nature discrète et dont les valeurs appartiennent à un ensemble fini. On trouve par exemple les commandes invoquées par un utilisateur.

b) Mesure continue : C'est une mesure réelle qui peut prendre des valeurs quelconques. Par exemple le nombre de lignes imprimées pendant la session ou la durée de la session.

IDES traite chaque enregistrement d'audit quand il arrive sur le système. Pour détecter des comportements anormaux pendant une session, alors que tous les paramètres de la session ne sont pas encore disponibles, IDES extrapole les valeurs et les compare au profil de l'utilisateur.

III.3.1. Structure de l'IDES

La structure d'IDES est donnée par la Figure 3.2. La donnée IDES consiste en données suivantes: Les données d'audit base consistent en un tableau de décryptage et de non-traitement valide.

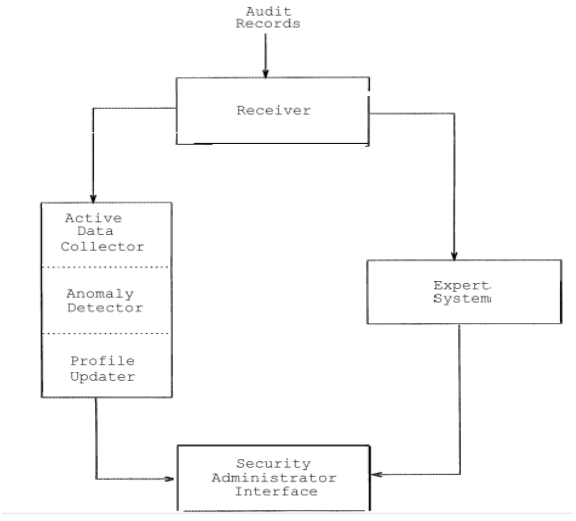

Figure 3.2. Structure de l'IDES

a) Les données d'audit : Consiste en une table contenant les journaux d'audit valides décryptés et non traités reçus à partir de la cible.

b) Les données actives : Enregistrent l'activité accumulée pour chaque utilisateur, groupe, hôte distant et le système dans son ensemble depuis la dernière mise à jour du fichier profile.

c) Les données archivées : Elles consistent en un tableau d'enregistrements d'audit traités et d'un ensemble de fichiers. Les journaux d'audit sont périodiquement déplacés vers un nouveau fichier et les fichiers sont régulièrement sauvegardés.

d) Les données de profile : Consiste en plusieurs tables qui définissent un comportement normal basé sur le comportement passé de chaque sujet. Les profils sont mis à jour quotidiennement à l'aide du comportement le plus récemment observé (à partir des données actives).

e) Les données d'anomalies : Consiste en un ensemble de tableaux contenant des enregistrements d'anomalie. Un journal d'anomalie est généré lorsque le comportement actuel observé s'écarte anormalement du comportement normal tel que spécifié dans les profils, ou lorsque les règles spécifiées dans le composant du système expert ont été violées.

III.4. GrIDS

GrIDS [21] est un système de détection d'intrusions pour de grands réseaux où l'activité réseau est représentée par un graphe. Les hôtes du réseau sont les nœuds du graphe et les connexions entre les hôtes sont les arêtes. Le responsable de la sécurité du réseau établit un ensemble de règles qui permet de décider quel trafic entre les hôtes va représenter l'activité entre ces hôtes. Le graphe et les arêtes sont libellés par des attributs qui sont données par les règles suscitées.

La construction de l'activité du réseau repose sur le paradigme organisationnel d'une hiérarchie de départements, et ce fait comme suit [19] : Un département est formé de plusieurs hôtes, et il centralise les données d'audit et les combine pour générer le graphe du département, en se référant à l'ensemble de règles spécifié. Si des événements réseau d'un hôte du département impliquent un hôte hors du département, alors les graphes des deux départements peuvent être combinés, selon des règles spécifiées. Le nouveau graphe se compose donc de deux sommets, qui sont les départements et d'une arête qui spécifie le trafic entre ces deux départements. Ce processus est répété, et une hiérarchie de départements est formée.

III.4.1. Architecture de GrIDS [21]

La Figure 3.3 représente une hiérarchie simple avec trois départements : Gauche avec trois hôtes, Droite avec un hôte et Père qui contient gauche et droite.

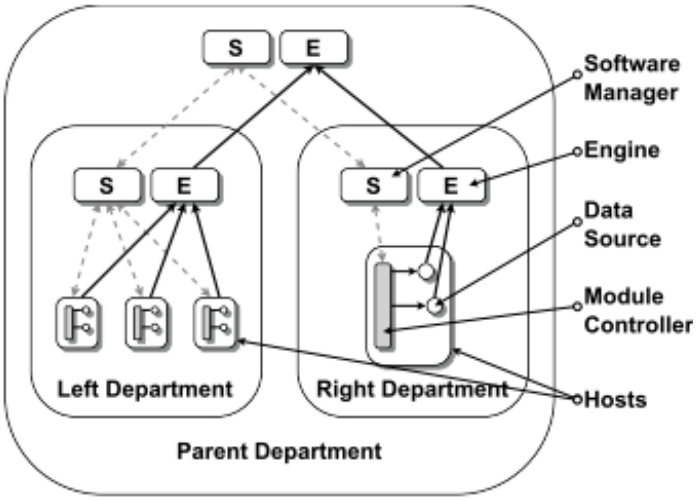

Figure 3.3. Architecture de GrIDS

Chaque département dispose de deux modules spéciaux : le gestionnaire de logiciels et le moteur graphique. Le gestionnaire de logiciel gérer l'état de la hiérarchie et les modules distribués. La hiérarchie est réorganisée dynamiquement par glisser-déposer dans une interface utilisateur, de même le démarrage et l'arrêt de modules particulier est également automatisé.

Les sources de données GRIDS sont des modules qui surveillent l'activité des hôtes et des réseaux et envoient des rapports d'activité détectée au moteur. L'activité est signalée sous la forme d'un nœud ou d'une arrête pour une éventuelle inclusion dans le graphique d'activité.

Conclusion

Dans ce chapitre, nous avons donné une liste des IDS existant dans la littérature, puis avons présenté quelques IDS en donnant leur architecture et leur principe de fonction.

Dans le chapitre suivant, nous présentons d'bord l'IDS Snort de manière détaillé, puis nous procédons à sa mise en œuvre sous Linux.

Introduction

Au cours du chapitre précédent, nous avons présenté quelques IDS existant dans la littérature en donnant leur architecture ainsi que leur principe de fonctionnement d'une manière générale.

Dans ce qui suit, nous allons présenter et mettre en œuvre l'une des solutions les plus populaires : l'IDS SNORT. Nous commencerons d'abord par une description générale de SNORT. Ensuite, nous passerons aux différentes manipulations de mise en œuvre de l'outil SNORT, et on finira par la présentation des résultats auxquels on a abouti.

IV.1. Présentation de SNORT

SNORT est un système de détection d'intrusion réseau (NIDS) open source, disponible sous licence GPL, fonctionnant sur les systèmes Windows et Linux, à l'origine écrit par Martin Roesch.

SNORT est un IDS capable d'effectuer l'analyse du trafic sur un réseau en temps réel et de la journalisation de paquets sur des réseaux IP. Il peut effectuer l'analyse des protocoles, la recherche / correspondance de contenu et peut être utilisé pour détecter une variété d'attaques listées dans sa base de données.

IV.1.1. Fonctionnement de SNORT:

SNORT fonctionne en quatre modes :

1. **Le mode sniffer :** dans ce mode, SNORT lit les paquets circulant sur le réseau et les affiche d'une façon continue sur l'écran.
2. **Le mode « packet logger » :** dans ce mode, SNORT journalise le trafic réseau dans des répertoires sur le disque.
3. **Le mode détecteur d'intrusion réseau (NIDS) :** dans ce mode, SNORT analyse le trafic du réseau, compare ce trafic à des règles déjà définies par l'utilisateur et établi des actions à exécuter.
4. **Le mode Prévention des intrusions réseau (IPS) :** le mode IPS n'est plus SNORT proprement parlé, Il s'agit d'une autre version appelée Snort inline, Cette version permet de modifier ou de rejeter des paquets.

IV.1.2. Positionnement de SNORT dans le réseau

L'emplacement de SNORT sur un réseau joue un rôle très important vis-à-vis de son efficacité.

Pour un réseau composé d'un firewall et d'une DMZ, trois positions sont envisageables, voir Figure IV.1:

Position1 : Si SNORT est placés avant le firewall, il 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 difficiles à consulter.

Position 2 : Si SNORT est placé sur la DMZ, il détectera las attaques qui n'ont pas été filtrées pas 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 : dans le cas d'un positionnement sur le réseau interne, SNORT peut rendre compte des attaques internes, provenant du réseau local. Il est judicieux d'en placer un à cet endroit du 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 afin de pouvoir ensuite les éradiquer.

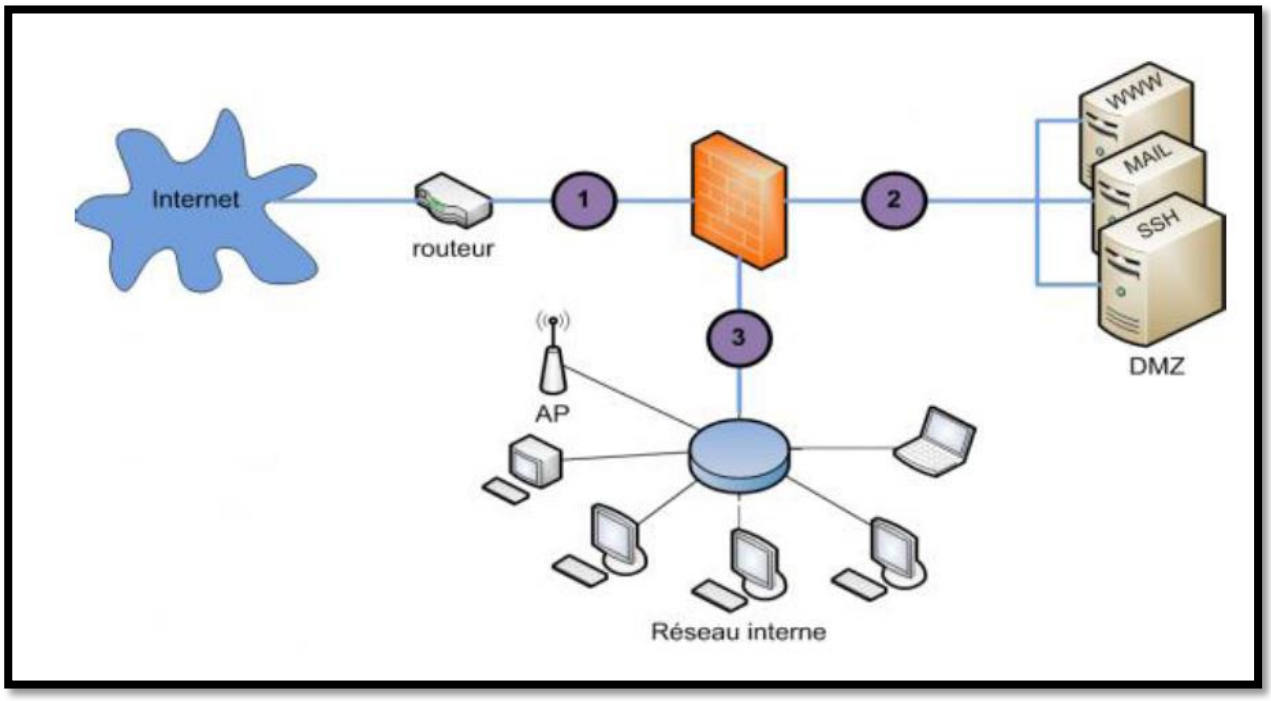

Figure VI.1 : Les différentes positions de SNORT dans un réseau

En ce qui est de notre travail, nous avons opté pour la position 1.

IV.1.3. Architecture de SNORT

L'architecture de SNORT est modulaire, elle est composée de :

1. Un noyau de base (Packet Decoder) : au démarrage, ce noyau charge un ensemble de règles, les compile, les optimise et les classe. Durant l'exécution, le rôle principal du noyau est la capture des paquets.

2. Une série de pré–processeurs (Detection Engine) : ces derniers améliorent les possibilités de SNORT en matière d'analyse et de recomposition du trafic capturé. Ils reçoivent les paquets directement capturés et décodés, les retravaillent éventuellement puis les fournissent au moteur de recherche des signatures pour les comparer avec la base des signatures.

3. Une série de « Detection plugins »: Ces analyses se composent principalement de comparaison entre les différents champs des headers des protocoles (IP, ICMP, TCP et UDP) par rapport à des valeurs précises.

4. Une série de « output plugins »: permet de traiter cette intrusion de plusieurs manières : envoie vers un fichier log, envoie d'un message d'alerte vers un serveur syslog, stocker cette intrusion dans une base de données SQL.

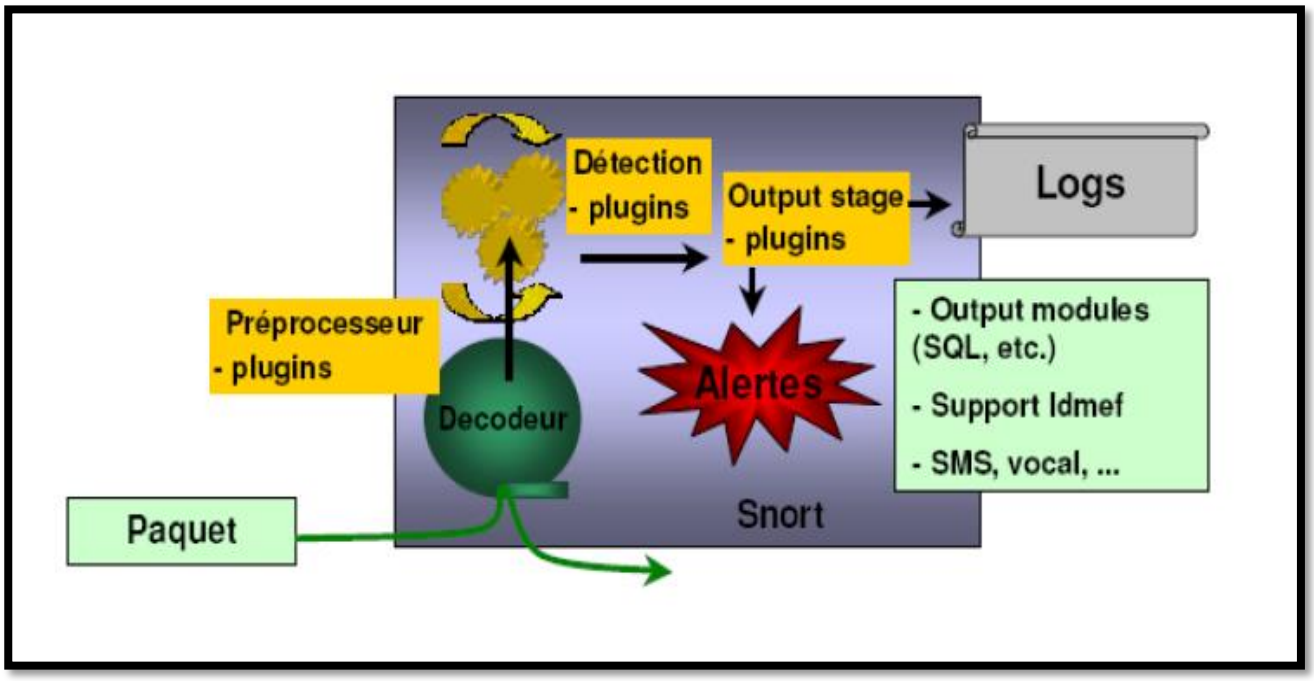

Figure IV.2 : Architecture de SNORT

IV.1.4. Environnement

Dans le souci d'atteindre une sécurité élevée, nous avons opté pour un environnement de travail Linux, plus précisément CentOS 6.5. Cet environnement nous offre un espace de travail favorable et nous assure une fiabilité des résultats.

CentOS est une distribution GNU/Linux principalement destinée aux serveurs. Elle est l'une des distributions Linux les plus populaires pour les serveurs web.

IV.2Paramétrage de Snort

IV.2.1. Préprocesseurs

Les préprocesseurs permettent d'étendre les fonctionnalités de SNORT. Ils sont exécutés avant le lancement du moteur de détection et après le décodage du paquet IP. Le paquet IP peut être modifié ou analysé de plusieurs manières en utilisant le mécanisme de préprocesseur.

Les préprocesseurs sont chargés et configurés avec le mot-clé préprocesseur. Le format de la directive préprocesseur dans les règles de SNORT est :

Préprocesseur<nom> : <options>.

IV.2.1.1. Mainfrag

Le préprocesseur mainfrag examine les paquets fragmentés pour une valeur limite spécifiée. Quand les paquets sont fragmentés, c'est généralement causé par des routeurs entre la source et la destination. Généralement, il n'y a pas d'équipement réseau commercial qui fragmente les paquets en tailles plus petits que 512 octets, donc nous pouvons utiliser ce fait pour activer la surveillance réseau pour des petits fragments qui sont généralement indicatifs de quelqu'un essayant de cacher son trafic derrière la fragmentation.

Format : *mainfrag :<valeur limite>*

Exemple *: preprocessor mainfrag : 128*

IV.2.1.2. HTTP Decode

HTTP Decode est utilisé pour traité les chaines HTTP URL et convertir leurs données en chaines ASCII non obscurcies. Ceci est fait pour vaincre les scanners évasifs d'URL web et les attaque hostiles qui pourraient autrement échappé aux chaines d'analyse de contenu utilisé pour examiner dans le trafic http des activités suspicieuses

Le module préprocesseur prend des numéros de port http (séparer par des espaces) pour être normaliser en arguments (typiquement 80 et 8080).

Format : *http_decode : <liste de port>*

Exemple : *preprocessor http_decode : 80 8080*

IV.2.1. 3. Détecteur de balayage du port (Portscan Detector)

Le préprocesseur de scans des ports et responsable de plusieurs taches qui sont :

- Journaliser le début et la fin du scan depuis une unique source IP pour faciliter la journalisation standard.
- Si un fichier journal est spécifié, il journalise les IP destination et ports scannés ainsi que le type du scan.
- Un scan de ports est défini comme des tentatives de connexions TCP ver plus de ports P et en secondes T, ou des paquets P UDP envoyés à plus de ports P en secondes T.

Format : *portscan<réseau surveillé><nombre de ports><période de détection><répertoire/fichier/journal.log>*

Exemple : *preprocessor portscan : 192.168.1.0/24 12 5 /var/log/portscan.log*

IV.2.1.4. Défragmentation(Defrag)

Le module defrag (de Dragos Ruiu) permet a Snort de faire de la défragmentation IP, rendant plus facile aux pirates de contourner simplement les capacités de détections du système. Il est plus très simple a utilisé, requérant simplement l'addition d'une directive préprocesseur au fichier de configuration sans aucun argument« supercedes ».

Format : *defrag*

Exemple : *preprocessor defrag*

IV.2.1.5. Stream

Le plugin Stream fournit la fonctionnalité de réassemblage de sessions TCP à Snort. Les sessions de TCP sur les ports configurés avec de petits segments réassemblés en un flux de données que Snort peut évaluer proprement pour des activités suspicieuses. Ce plugin prend plusieurs arguments :

- Timeout : temps maximal en secondes pendant laquelle une session sera gardée vivante s'elle na pas eu un paquet pour elle.
- Port : le port serveur a surveillé (session TCP a surveillé).
- Maxbytes : le nombre maximal d'octets dans notre paquet reconstruit.

Format : *stream: timeout <temps maximal>, port <ports>, maxbytes <maximum d'octets>*

Exemple :*preprocessor stream timeout 5, port 5 7 80, maxbytes <157962>*

IV.2.2. Les plugin de sortie

Les modules de sortie sont une nouveauté de la version 1.6. Ils permettent a SNORT d'être bien plus flexible dans le formatage et la présentation des sorties à ses utilisateurs. Les modules de sortie sont exécutés quand les sous-systèmes d'alerte ou de journalisation de SNORT sont appelés. Après les préprocesseurs et le moteur de détection. Le format des directives dans le fichier des règles est très similaire à celui des préprocesseurs.

Plusieurs plugins de sortie peuvent êtres spécifiés dans le fichier de configuration de SNORT. Quand plusieurs plugins du même type (journal, alerte) sont spécifiés, ils sont empilés et appelés en séquence quand un évènement se produit. Comme avec les systèmes standards de journalisation et d'alertes, les plugins de sortie envoient leurs données a /var/log/snort par défaut ou vers un répertoire désigné par un utilisateur (en utilisant l'option de la ligne de commande « -1 »).

IV.2.2.1.Alerte syslog (Alert_syslog)

Ce module envoie les alertes à la *facilité syslog* (comme l'option –s de la ligne de commande). Ce permet également à l'utilisateur de spécifier la *facilité de journalisation* et la priorité dans le fichier de règles de Snort, en donnant aux utilisateurs une plus grande flexibilité dans la journalisation des alertes.

Les mots clés disponibles sont :

Options:

LOG_CONS, LOG_NDELAY, LOG_PERROR, LOG_PID, LOG_AUTH, LOG_AUTHPRIV, LOG_DAEMON, LOG_LOCAL, LOG_USER

Priorités:

LOG_EMERG, LOG_ALERT, LOG_CRIT, LOG_ERR, LOG_WARNING, LOG_NOTICE, LOG_INFO, LOG_DEBUG

IV.2.2.2. Alerte rapide (alert_fast)

Ceci imprimera les alertes de SNORT dans un format rapide d'une ligne vers un fichier de sortie spécifié. C'est une méthode d'alerte plus rapide que les alertes *full* car elle n'a pas besoin d'afficher toutes les entêtes des paquets vers le fichier de sortie.

Format : *alert_fast :<nom du fichier de sortie>*

Exemple : *output_alert_fast :alert.fast*

IV.2.2.3. Alerte pleine (Alert_full)

Affiche les messages d'alerte de SNORT avec l'intégralité des entêtes des paquets. Cette facilité d'alerte est généralement plutôt lente car elle requière que le programme fasse beaucoup plus d'analyse de données pour formater les données à être imprimées. Les alertes seront écrites dans le répertoire de journalisation par défaut (/var/log/snort) ou le répertoire de journalisation spécifié sur la ligne de commande.

Format : *alert_full :<nom du fichier de sortie>*

Exemple: *output_alert_full :alert.full*

IV.2.2.4. Alerte smb

Ce plugin envoie des messages d'alerte Winpopup aux noms de machines NETBIOS indiqués dans le fichier spécifié comme argument à ce plugin de sortie.

Il est à noter que l'utilisation de ce plugin n'est pas encouragée puisqu'il exécute un exécutable binaire externe (smbclient) au même niveau de privilège que SNORT, communément *root.* Le format du fichier des stations de travail est une liste de noms NETBIOS des systèmes qui souhaitent recevoir les alertes, par une ligne de ce fichier.

Format : *aletr_smb :<nom du fichier des stations de travail à alerter>*

Exemple: *output alert_smb : workstation.list*

IV.2.2.5. Alerte unixsock

Configure un socket du domaine UNIX et y envoie les rapports d'alerte. Des programmes / processus externes peuvent écouter cette socket et recevoir les alertes Snort et les données des paquets en temps réel.

Format : *alert_unixsock*

Exemple : *output alert_unixsock*

IV.2.2.6. Log_tcpdump

Le module log_tcpdump enregistre les parquets vers un fichier au format tcpdump. Ceci est utile pour effectuer des analyses post traitement sur le trafic collecté avec le grand nombre d'outils qui sont disponible pour examiner des fichiers au format tcpdump. Ce module ne prend qu'un seul argument, le nom du fichier de sortie.

Format : *log_tcpdump :<nom du fichier de sortie>*

Exemple : output log_tcpdump : snort.log

IV.2.3. Les bases de Snort

Snort permet d'écrire des règles personnelles et utilise un langage simple et léger de description de règles qui est flexible et assez puissant.

Les règles de Snort sont devisées en deux sections logiques, l'entête et les options de la règle :

- L'entête de la règle : il contient comme information l'action de la règle, le protocole, les adresse IP source et destination, les masques réseaux et les ports source et destination.
- Option de la règle : contient les messages d'alertes et les informations sur les parties du paquet qui doivent être inspectées pour déterminer si l'action de la règle doit être acceptée.

Exemple : *alert tcp any any -> 192.168.1.0/24 (content : ''|00 01 86 a5|'' ; msg ; '' mountd access '' ;)*

IV.2.3.1. Les inclusions

Le mot clé *include* permet à d'autres fichiers de règles d'être inclus dans le fichier de règles indiqué sur la ligne de commande de SNORT. Il fonctionne comme un *#include* du langage de programmation C, il lit le contenu des fichiers nommés et les met en place dans le fichier à la place où l'*include* apparait.

Format : *include :<répertoire/nom du fichier include>*

Exemple: *include $RULE_PATH/emerging-botcc-BLOCK.rules*

IV.2.3.2. Les variables

Des variables peuvent être définies dans SNORT. Ce sont de simples substitutions des variables fixées avec le mot clé *var*.

Format: *var :<nom> <valeur>*

Exemple: var MY_NET [192.168.1.0/24,10.1.1.0/24] alert tcp any any -> $MY_NET any (flags:S;msg: "SYN packet";)

IV.2.4. Les règles de Snort

IV.2.4.1. Création des règles :

Les règles de SNORT sont composées de deux parties distinctes : le header et les options.

- 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 [18].

Action	Protocole	Adresse1	Port1	Direction	Adresse2	Port2	Options (msg, content..)

Figure IV.3 : Format des règles de Snort.

IV.2.4.1.1. Header

- **Le champ « action » :** il peut prendre plusieurs valeurs selon l'action à mener par Snort en détectant des paquets réseaux répondant au critère défini dans la règles. Ces valeurs sont les suivantes :
  -alert : génère une alerte et log le paquet.
  -log : log le paquet
  -pass : ignore le paquet
  -activate : active une règle dynamique
  -dynamic : définie une règle dynamique.
  -…etc.
- **Le champ « Protocole » :** décrit le protocole utilisé pour la communication. Snort supporte les protocoles TCP, UDP, ICMP et IP.
- **Les champs « Direction » :** renseignent Snort sur la direction des échanges réseau (->, <-, <->).
- **Les champs «Adress/Port » :** décrivent les adresses IP et les ports des machines qui échangent des données sur le réseau.

IV.2.4.1.2. Options

Pour chaque option, le format est : nom (option). Ci-dessous les options utilisées dans la création des règles :

- **msg** : affiche un message dans les alertes et journalise les paquets.
- **Logto** : journalise le paquet dans un fichier nommé par l'utilisateur au lieu de la sortie standard.
- **Ttl** : teste la valeur du champ TTL de l'entête IP.
- **Tos** : teste la valeur du champ TOS de l'entête.
- **Id** : teste le champ ID de fragment de l'entête IP pour une valeur spécifiée.
- **Ipoption** : regarde les champs des options IP pour des codes spécifiques

- **Fragbits** : teste les bits de fragmentation de l'entête IP.
- **Dsize** : teste la taille de la charge du paquet contre une valeur.
- **Flags** : teste les drapeaux TCP pour certaines valeurs.
- **Seq** : teste le champ TCP de numéro de séquence pour une valeur spécifique.
- **Ack** : teste le champ TCP d'acquittement pour une valeur spécifiée.
- **Itype** : teste le champ type ICMP contre une valeur spécifiée.
- **Icode** : teste le champ code ICMP contre
- **Icmp_id** : teste le champ ICMP ECHO ID contre une valeur spécifiée.
- **Icmp_seq** : teste le numéro de séquence ECHO ICMP contre une valeur spécifiée.
- **Content** : recherche un motif dans la charge d'un paquet.
- **Content-list** : recherche un ensemble de motifs dans la charge d'un paquet.
- **Offset** : modifie l'option contente, fixe le décalage du début de la tentative de correspondance de motif.
- **Depth** : modifie l'option content, fixe la profondeur maximale de recherche pour la tentative de correspondance de motif.
- **Nocase** : correspond à la procédure de chaine de contenu sans sensibilité aux différences majuscules/minuscules
- **Session** : affiche l'information de la couche applicative pour la session donnée.
- **Rpc** : regarde les services RPC pour des appels à des applications/procédures spécifiques.
- **Resp** : réponse active (ex: ferme les connexions).
- **React** : réponse active (bloque les sites web)

Mise à jour des règles

Les mises à jour des règles de Snort sont disponibles sur le site officiel http://www.snort.org. Cependant, une inscription annuelle est requise.

IV.3. Barnyard2

Barnyard permet de prendre en charge l'inscription des événements en base de données et libère donc des ressources à Snort qui peut davantage se concentrer sur la détection des intrusions, ainsi Snort inscrira les événements dans des logs au format unifié (FastUnifiedLogging) et ses derniers seront exploités par Barnyard pour une inscription en base de données.

IV.3.1. Le plugin « unified2 »

Le plugin de sortie unifié est conçu pour être la méthode la plus rapide possible de la journalisation des événements de Snort. Il enregistre les événements dans un format binaire ce qui permet encore d'alléger le mécanisme sur les alertes des événements.

Le nom unifié est un terme impropre, puisque le plugin de sortie unifié crée deux fichiers différents, un fichier d'alerte et un fichier journal.

- Le fichier d'alerte : contient les détails de haut niveau d'un événement (par exemple : IP, Protocole, Port, identifiant de message).
- Le fichier journal : contient les informations de paquets détaillés (un dump paquet avec l'ID d'événement associé).

Les deux types de fichiers sont écrits dans un format binaire.

IV.4. La console B.A.S.E :

Par défaut, les alertes de Snort sont enregistrées dans un simple fichier texte. L'analyse de ce fichier n'est pas aisée, même en utilisant des outils de filtre et de tri. C'est pour cette raison qu'il est vivement conseillé d'utiliser des outils de monitoring. Parmi ceux-ci, le plus en vogue actuellement est B.A.S.E (Basic Analysis and Security Engine), un projet open-source basé sur ACID (Analysis Console for Instruction Databases).

La console BASE est une application Webérite en PHP qui interface la base de données dans laquelle Snort stocke ses alertes. Pour fonctionner, B.A.S.E a besoin d'un certain nombre de dépendances :

- Un SGBD installé, par exemple MySQL.
- Snort compilé avec le support de ce SGBD.
- Un serveur http, par exemple Apache.
- PHP5 : module PHP.
- PHP-MySQL : interface PHP/MYSQL.
- La bibliothèque ADODB (Active Data Object Data Base), destinée à communiquer avec différents systèmes de gestion de base de données (SGBD) comme MySQL, SQL server, …etc. Ecrite au début en PHP, il existe également une version en Python.
- PHP-Mail : extension PHP.

IV.5. Installation de Snort

Avant de lancer l'initialisation de SNORT sous CentOS, nous avons besoin d'installer quelques pré-requis de compilation ainsi que les dépendances de Snort qui sont :

- GCC (GNU Compiler Collection) : compilateur du C, C++, Java …sous Linux.
- Libpcap (Packet Capture Library) : Librairie utilisée pour capturer les paquets.
- Libdnet : bibliothèque logicielle open source permettant de fabriquer et d'injecter facilement des paquets sur le réseau.
- Libpcre : Librairie de fonction utilisant la même syntaxe et sémantique que perl 5.
- Daq : Permet d'acquérir des paquets sur le réseau. Indispensable pour les versions ultérieures à Snort 2.9.0.
- Zlib : bibliothèque logiciels de compression de donnée.

Une fois les éléments précédents sont installés, on peut lancer l'installation de Snort, cette dernière peut être décomposée en deux parties :

IV.5.1. L'installation de l'outil Snort

- Création d'un répertoire pour enregistrer les fichiers téléchargés

  Mkdir /usr/local/snort

- Accédé au répertoire :

  Cd /usr/local/snort/

- Télécharger l'outil Snort avec et la bibliothèque DAQ.

  Wget hhtp://www.snort.org/snort/snort-2.9.7.2.tar.gz

  Wget hhtp://www.snort.org/snort/daq-2.0.4.tar.gz

- Décompression de la bibliothèque DAQ

Tar –xvzf daq-2.0.4..tar.gz

- La Configuration du fichier décompressé

  ./configure

- La compilation

  Make

- L'Installation

  Make install

- Lancement de l'installation de l'outil Snort

  Tar –xvzf snort-2.9.7.2.tar.gz

  ./Configure

  make

  make install

IV.5.2.L'installation des regles Snort

- Copier le fichier de configuration snort.conf dans /etc/snort

  *cp * /usr/local/snort/snort.con /etc/snort/*

- Mise en place des règles snort dans le répertoire de configuration de snort

  cp snortrules-snapshot-2970.tar.gz /etc/snort/

- Accéder au répertoire /etc/snort

  cd /etc/snort

- Décompactage des règles

  tar –xvzf snortrules-snapshot-2970.tar.gz

IV.6. Lancement de snort

La vérification de l'installation de se fait en tapant la commande :

Snort –V

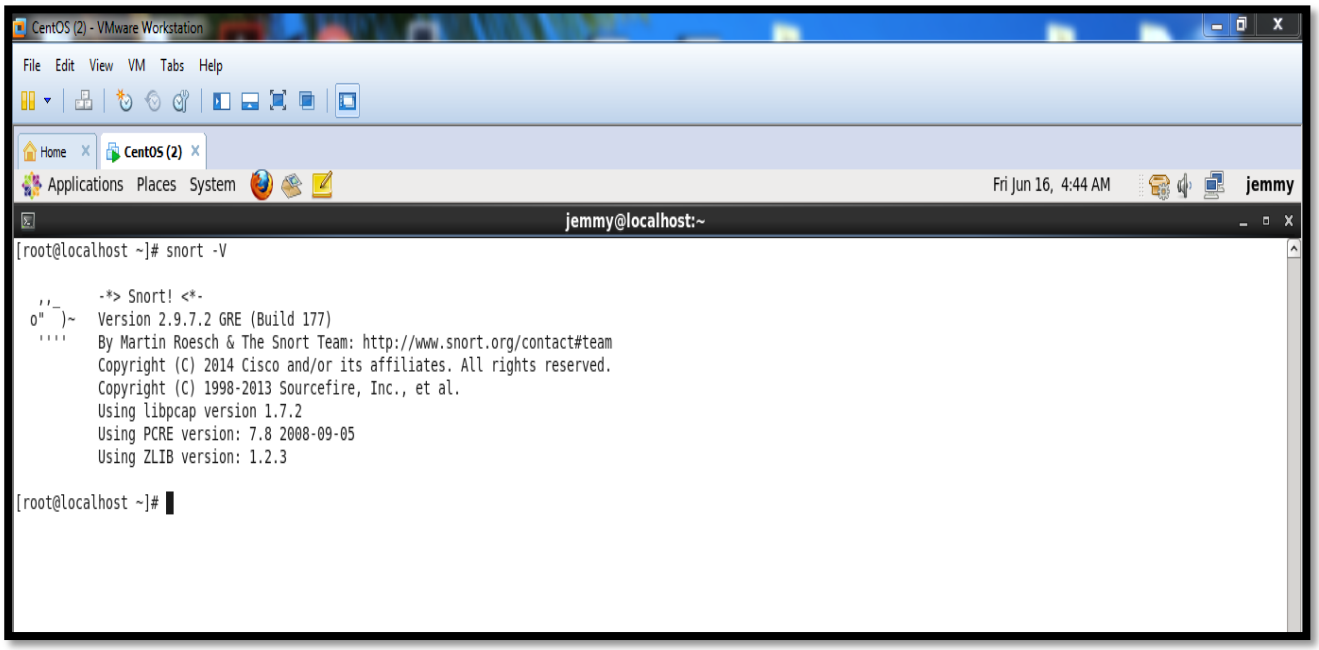

Figure IV.4 : Verification de l'installation de Snort.

IV.6.1. Les modes de fonctionnement

IV.6.1.1. Le mode ecoute (sniffer mode)

C'est le mode basic, il permet de lire et d'afficher les paquets TCP/IP circulant sur la reseau, d'une façon continue. On le lance avec la commande suivante :

Snort –vde

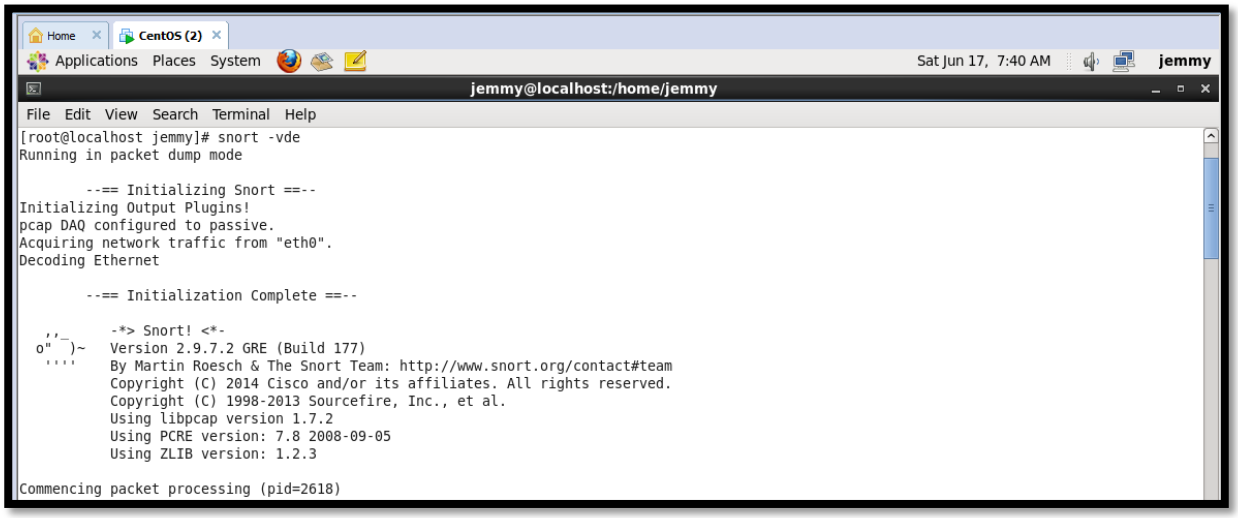

Figure IV.5 : Lancement du mode sniffer de Snort.

Capture sur la commande snort -vde

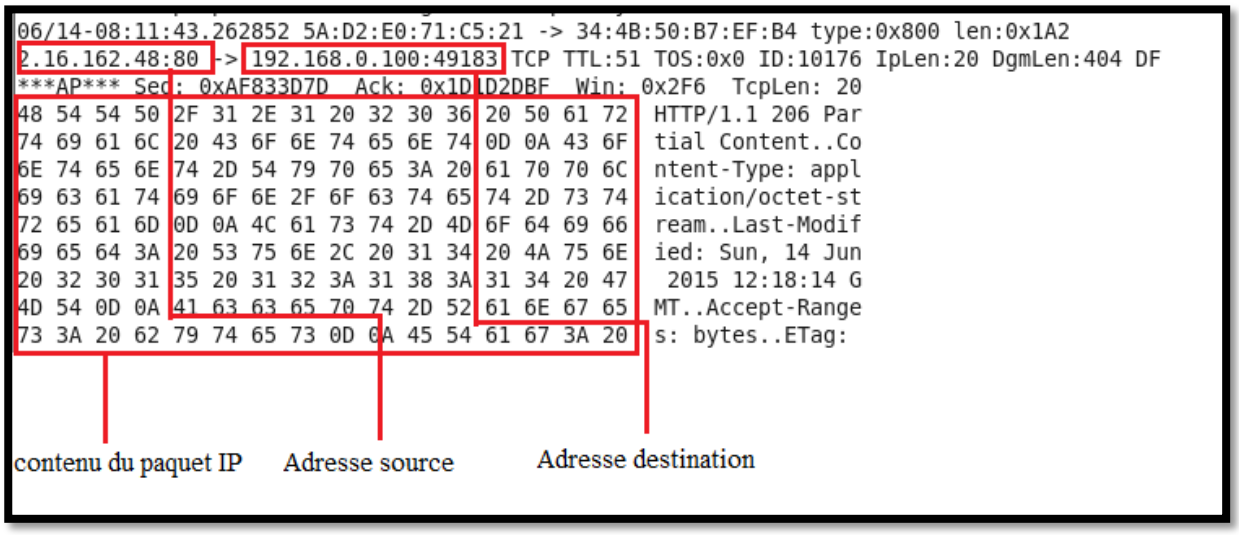

Figure IV.6 : Format d'un paquet analysé en mode sniffer.

IV.6.1.2. Le mode NIDS

Dans ce mode, SNORT analyse le trafic du réseau et le compare à des règles déjà définies par l'utilisateur et établit des actions à exécuter.

Pour faire, il faut d'abord installer tous les pré-requis, puis configurer le fichier snort.conf

Vi /etc/snort/snort.conf

Figure IV.7 : Spécification de l'adresse réseau à surveiller.

On doit indiquer l'adresse du réseau à surveiller comme sur l'image précédente

Il faudra aussi spécifier le répertoire où se trouvent les règles.

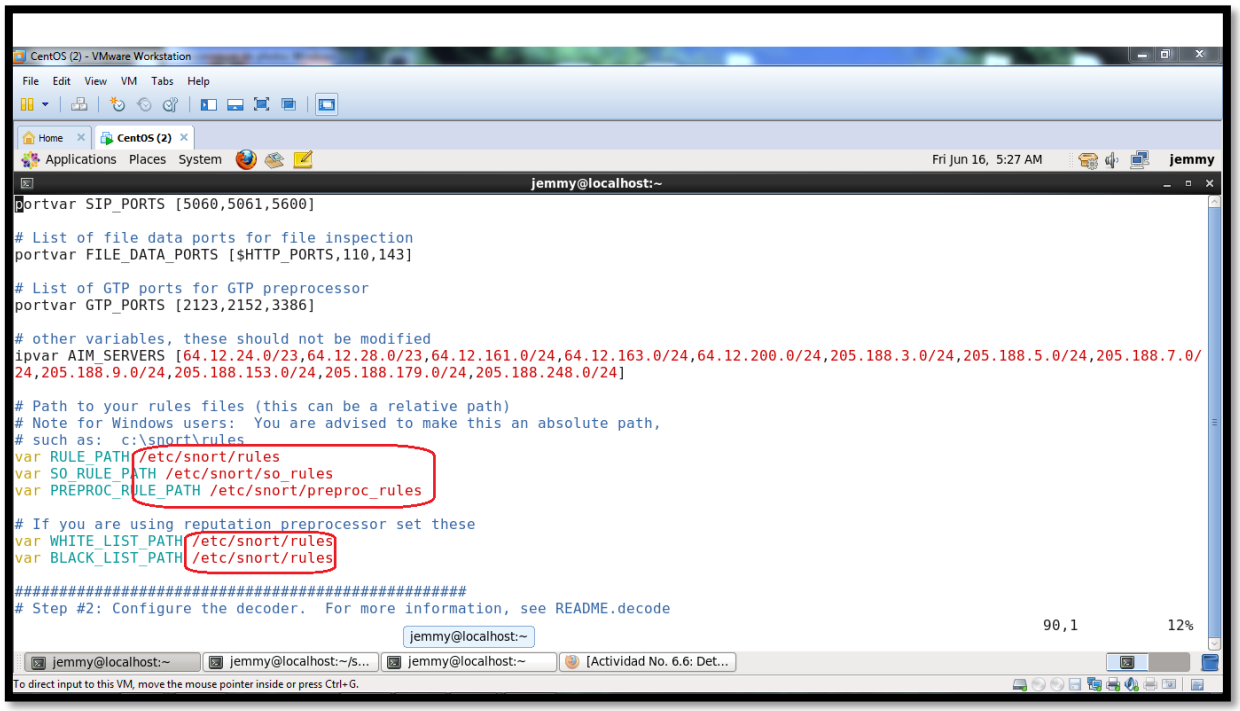

Figure IV.8 : Spécification du répertoire contenant les règles de Snort

On met # sur la ligne du fichier de sortie output log_tcpdump puisque nous avons opté pour le module de sortie unified.

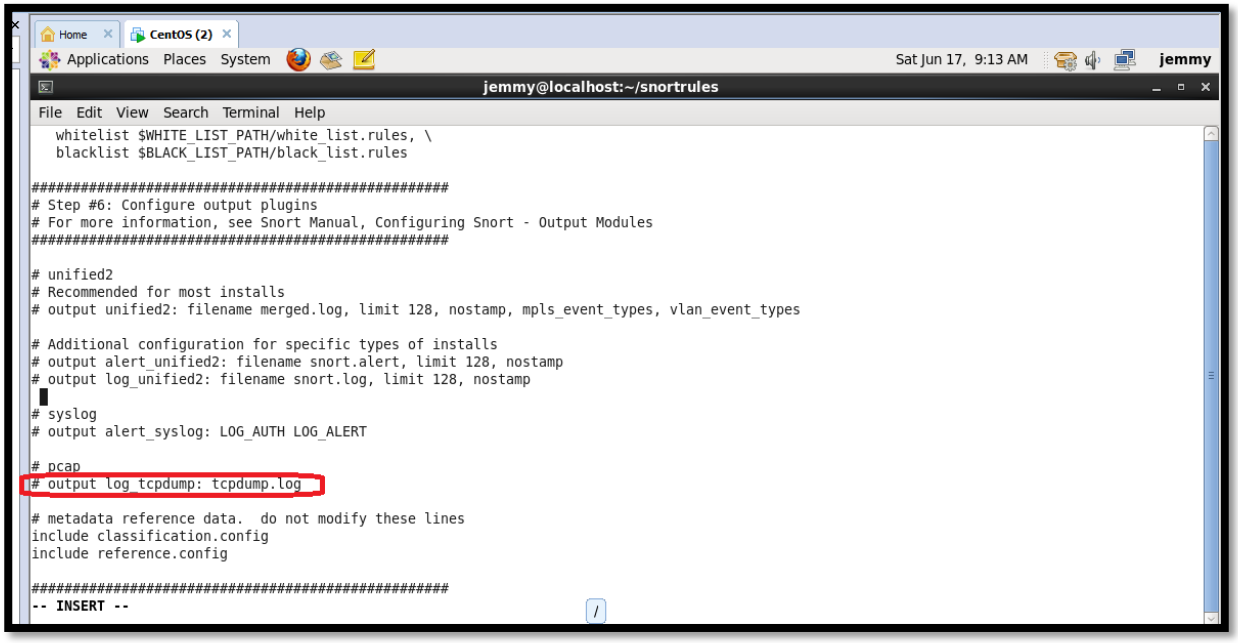

Figure IV.9 : Annulation de la méthode de sortie par défaut tcpdump.

Puis il faudra rajouter la ligne suivante : output unified2 : filename snort.log, limit 128

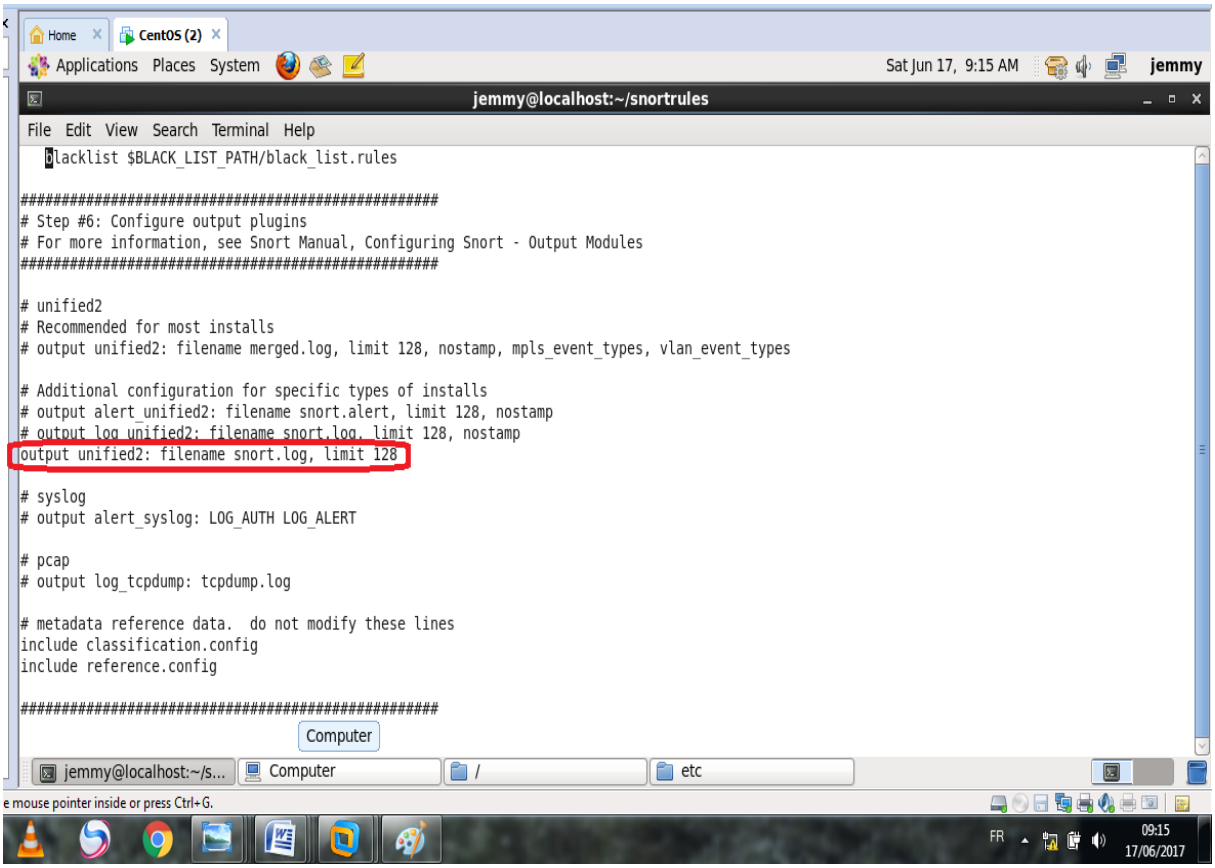

Figure IV.10 : Spécification de la méthode de sortie unified2.

IV.7. Mise en place de Barnyard2

IV.7.1. Installation de Barnyard2

On peut télécharger le package en exécutant la commande

Wget https://www.github.com/firnsy/barnyard2/archive/v2-1.13.tar.gz

Et ensuite l'installer en exécutant la suite de commandes suivante :

tar zxvf v2-1.13.tar.gz

cd barnyard2-2.1.13

autoreconf –fvi –I ./m4

./configure –with-mysql

make

make install

Ensuite, on copie le le fichier barnyard2.conf vers le répertoire /etc/snort afin de paramétrer Snort avec barnyard2 :

 # cp etc/barnyard2.conf /etc/snort

Enfin, on crée un dossier ou Barnyard2 stocke les logs :

 # mkdir /var/log/barnyard2

Le lancement ce fait avec la commande suivante :

Barnyard2 –c /etc/snort/barnyard2.con –d /var/log/snort –f snort.log –w /etc/snort/bylog.waldo /etc/snort/gen-msg.map

Figure IV.11 : Vérification de l'installation de barnyard2.

IV.7.2. Configuration du fichier baryard2.conf

Nous devons ajouter le nom du hôte 'localhost' est l'interface 'eth0' comme suit :

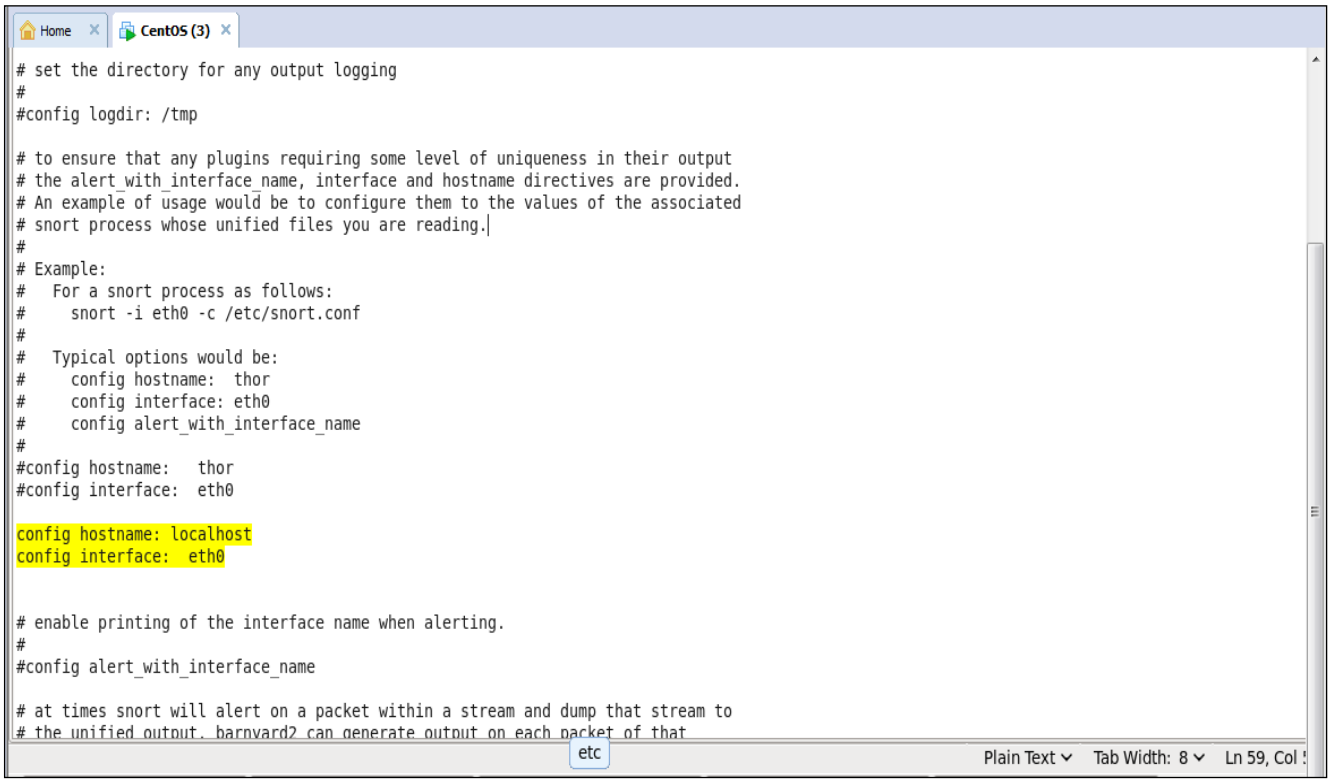

Figure IV.12 : Configuration du fichier baryard2.conf.

Ensuite nous devons configurer la sortie vers la base de données MySql

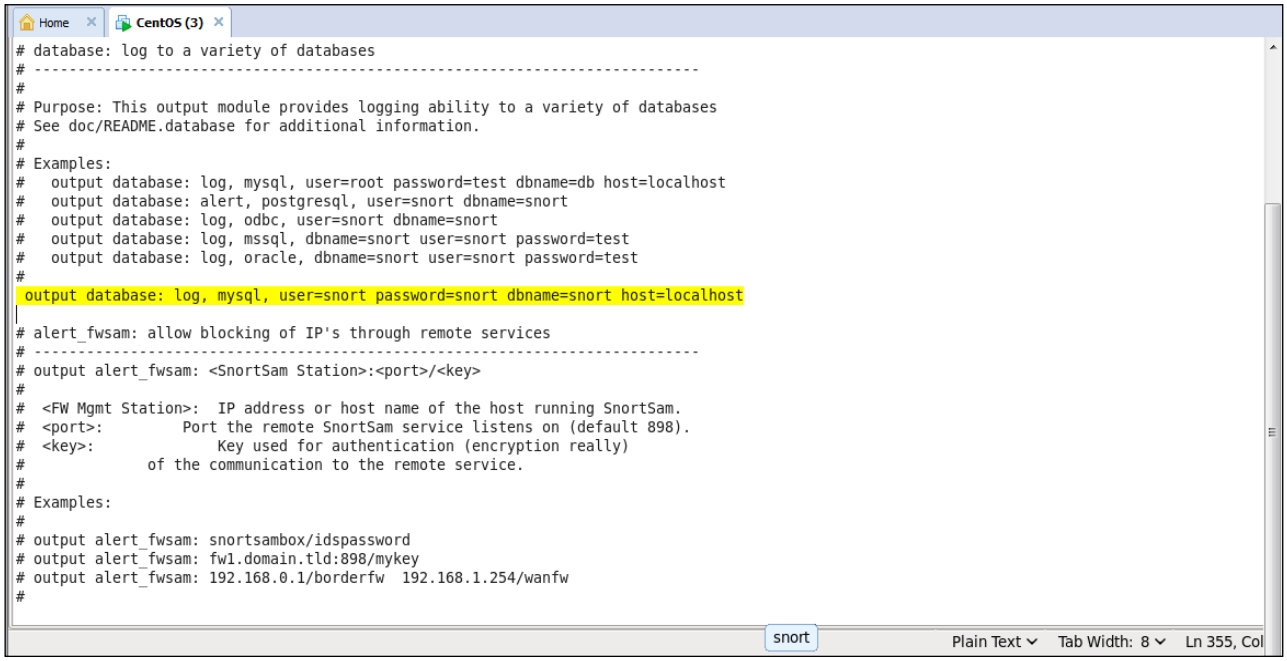

Figure IV.13 : Configuration de la sorite vers la base de données

Cette étape a pour but de synchroniser Snort avec Barnyard : On crée un fichier portant le nom de barnyard.waldo dans le répertoire : /var/log/snort

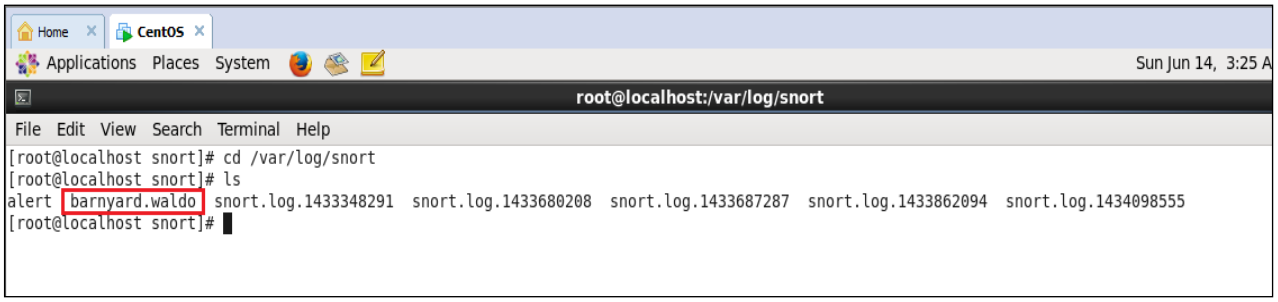

Figure IV.14 : Création du fichier Barnyard.waldo.

On tape la commande *#ls –la /var/log/snort* pour récupérer le dernier timestamp (le dernier en date).

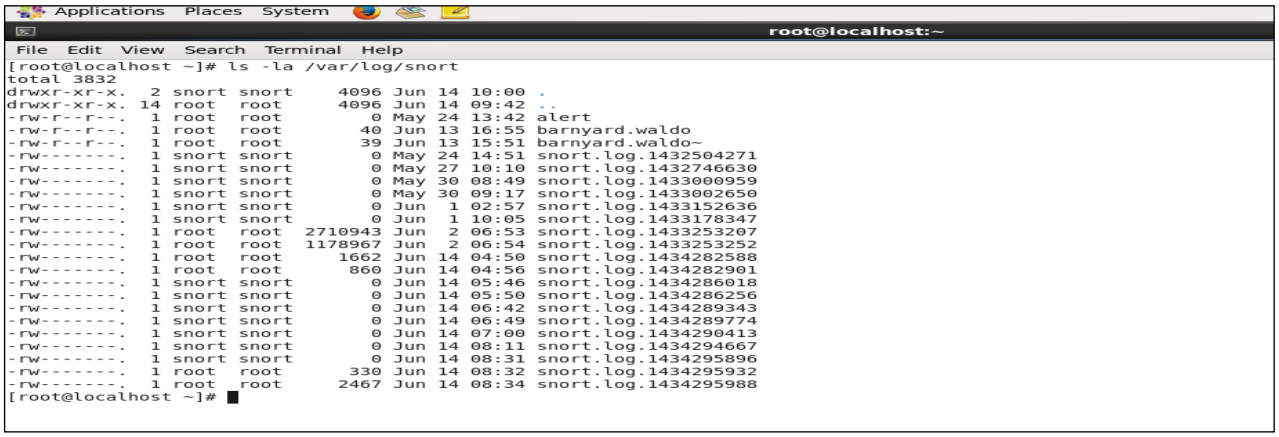

Figure IV.15 : Récupération du dernier timestamp.

Puis on l'injecte dans le fichier barnyard.walo

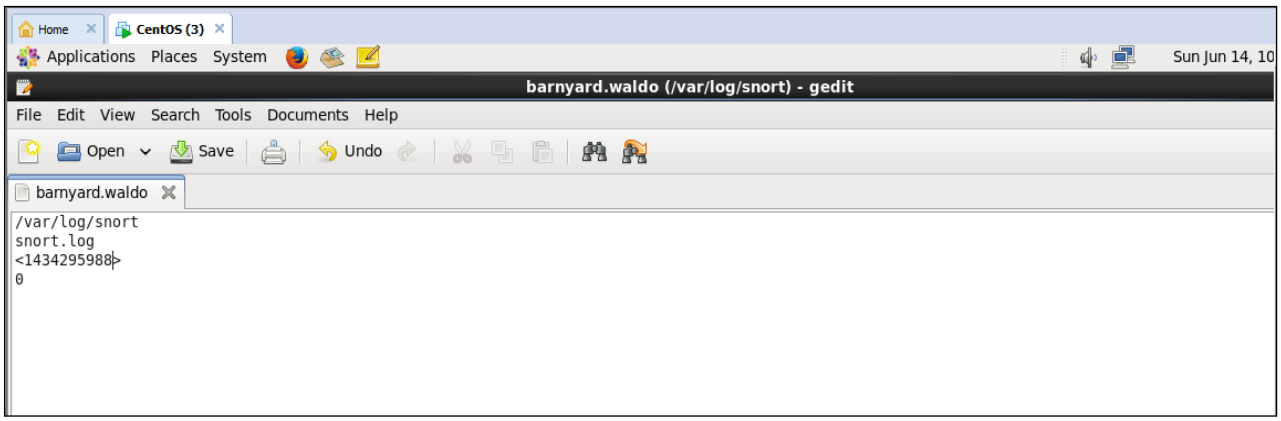

Figure IV.16: Injection du timestamp dans Barnyad.waldo.

VI.8. Mise en place de la base de données MySQL :

IV.8.1. Installation

La première étape consiste à installer MySQL-server et MySQL-devel :

yum install mysql-server mysql-devel

On doit démarrer ensuite le service mysql avec les deux commandes suivantes:

Chkconfig mysqld –add
service mysqld start

Nous pouvons accéder ensuite à la base de données avec la commande :

Mysql –u root –p

Comme le montre la figure suivante :

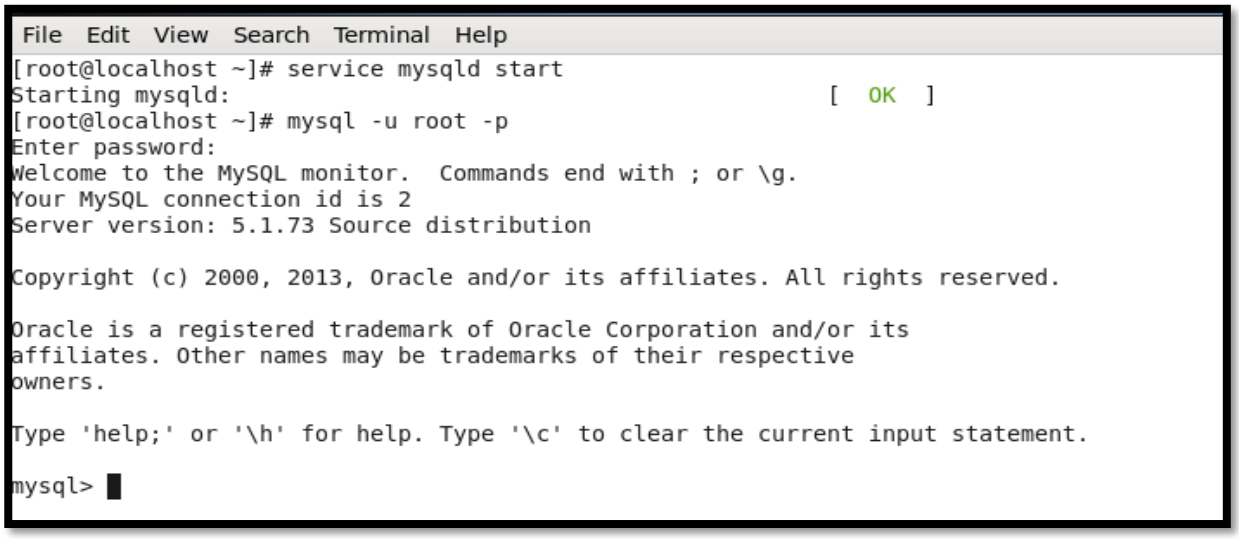

Figure IV.17 : Lancement du service MySQL.

IV.8.2. Création de la base de données pour snort

On lance d'abord MySQL, puis on crée la base de données Snort :

Mysql>create database snort;

Il est nécessaire de créer un utilisateur avec des permissions sur la base de données snort uniquement:

Mysql> grant all on snort. to snort@localhost;*

Mysql>set password for snort@localhost=password('snort');

Puis on recharge les privilèges MySQL:

Mysql> flush privileges ;

Mysql> exit

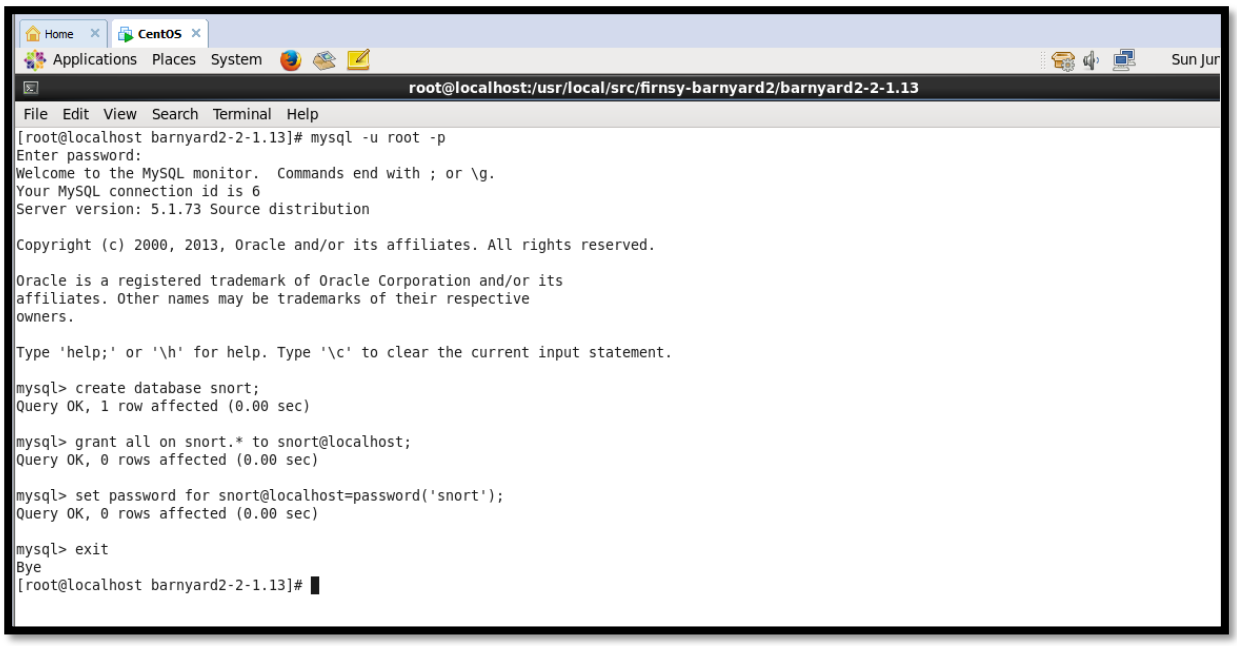

Figure IV.18 : Création de la base de données.

IV.8.3. Installation de snortmysql

Nous pouvons vérifier si la base snort à bien était créée par la commande :

Mysql> show database;

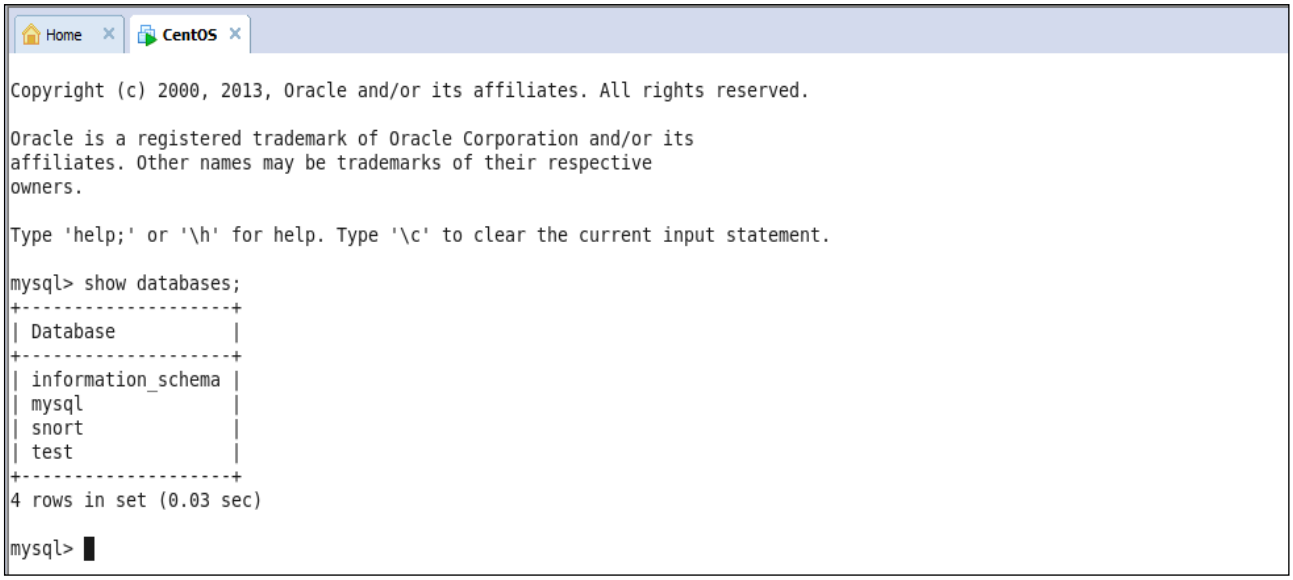

Figure IV.19 : Vérification de la création de la base de données.

Une fois la base de données créée, nous allons procéder à la création du schéma des données pour la base snort avec la commande :

Source /usr/local/src/create_mysql<- from barnyard2

Si tout va bien on aura la table suivante :

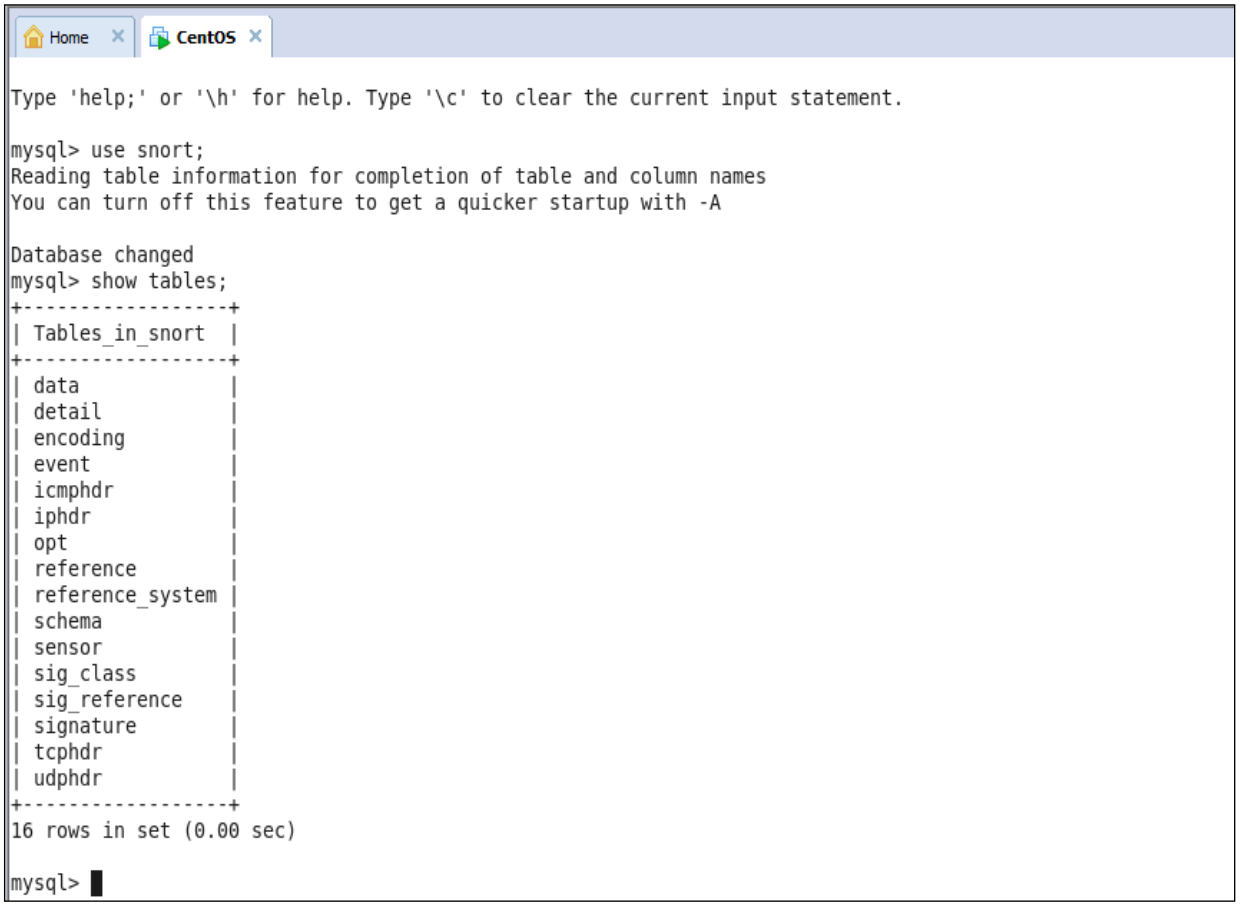

Figure IV.20 : Schématisation de la base de données.

VI.9. Mise en place de la console B.A.S.E

IV.9.1. Installation des pré-requis

yum install apache2 php5 libapache2-mod-php5 php5-gd php5-mysql libtool libpcre3-dev php-pear vim sshopenssh-server.

IV.9.2. Configuration du fichier php.ini

On configure le fichier php.ini pour que les modifications nécessaires soient apportées à PHP en ajoutant les extensions suivantes:

- *Extention=mysql.so*
- *Extension=gd.so*

Et en remplaçant la valeur de **Error_reporting** comme suit :

Error_reporting = E_ALL & ~E_NOTICE

Maintenant, nous pouvons démarrer le service http avec la commande :

service httpdstart

Figure IV.21 : Lancement du service http.

IV.9.3. Installation de B.A.S.E

On décompresse l'archive BASE

tar –xvzf base-1.4.5.tar.gz

Puis, on déplace le dossier base dans le répertoire /var/www/html/base avec la commande

Cp base-1.4.5 /var/html/base

IV.9.4. Installation d'Adodb

On décompresse le fichier adodb511.tgz avec la commande :

tarxvzf adodb511.tgz

```
[root@localhost /]# ls
adodb511.tgz       bin   etc              Image_Graph-0.8.0.tgz  Log-1.12.8.tgz  mnt         PEAR-1.9.5      pulledpork-0.7.0         sbin
barnyard           boot  home             lib                    lost+found      opt         PEAR-1.9.5.tgz  pulledpork-0.7.0.tar.gz  seli
base-1.4.5.tar.gz  dev   Image_Graph-0.8.0  Log-1.12.8           media           package.xml proc            root                     srv
[root@localhost /]# tar xvzf adodb511.tgz
adodb5/adodb-active-record.inc.php
adodb5/adodb-active-recordx.inc.php
adodb5/adodb-csvlib.inc.php
adodb5/adodb-datadict.inc.php
adodb5/adodb-error.inc.php
adodb5/adodb-errorhandler.inc.php
adodb5/adodb-errorpear.inc.php
adodb5/adodb-exceptions.inc.php
adodb5/adodb-iterator.inc.php
adodb5/adodb-lib.inc.php
adodb5/adodb-memcache.lib.inc.php
adodb5/adodb-pager.inc.php
adodb5/adodb-pear.inc.php
adodb5/adodb-perf.inc.php
adodb5/adodb-php4.inc.php
adodb5/adodb-time.inc.php
adodb5/adodb-xmlschema.inc.php
adodb5/adodb-xmlschema03.inc.php
```

Figure IV.22 : Installation de la bibliothèque ADODB.

Puis on le déplace dans le répertoire /var/www/base avec la commande

mv base /var/www/base

Après l'installation de B.A.S.E, on lance dans le navigateur : 127.0.0.1/base/ on aura l'interface de notre base :

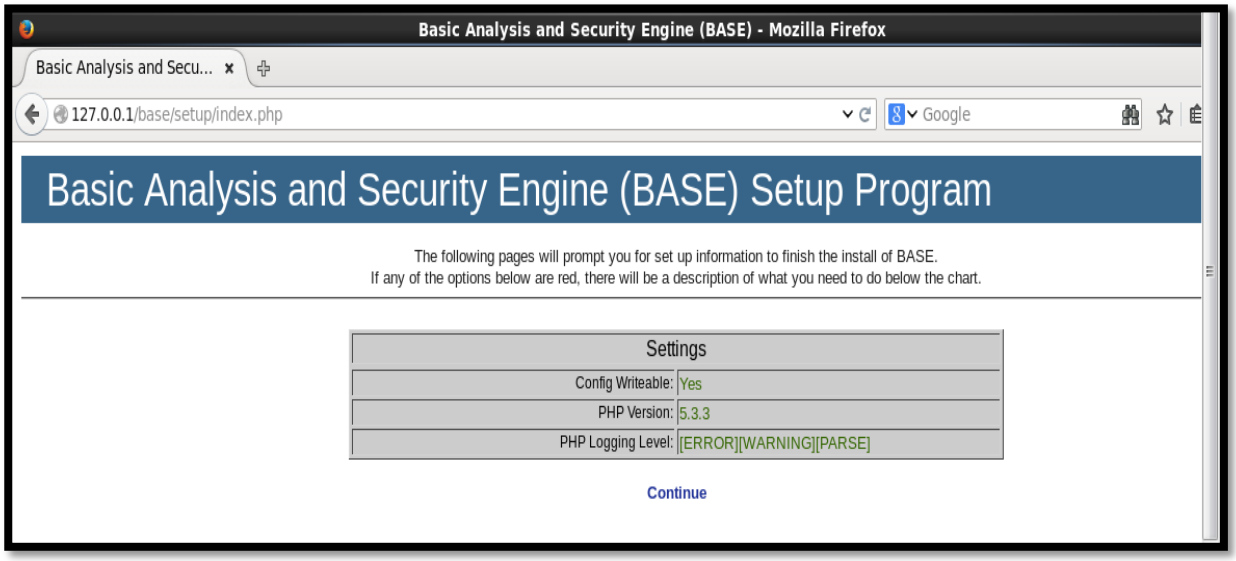

Figure IV.23 : Page d'accueil de B.A.S.E.

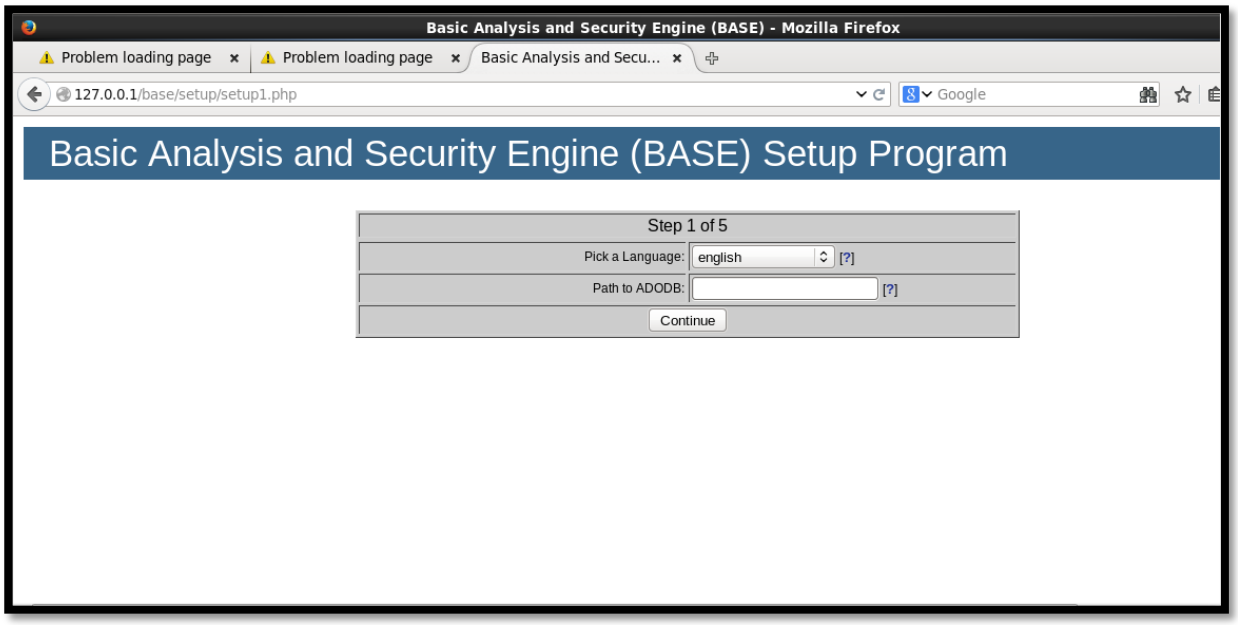

Figure IV.24 : Sélection de la langue pour B.A.S.E.

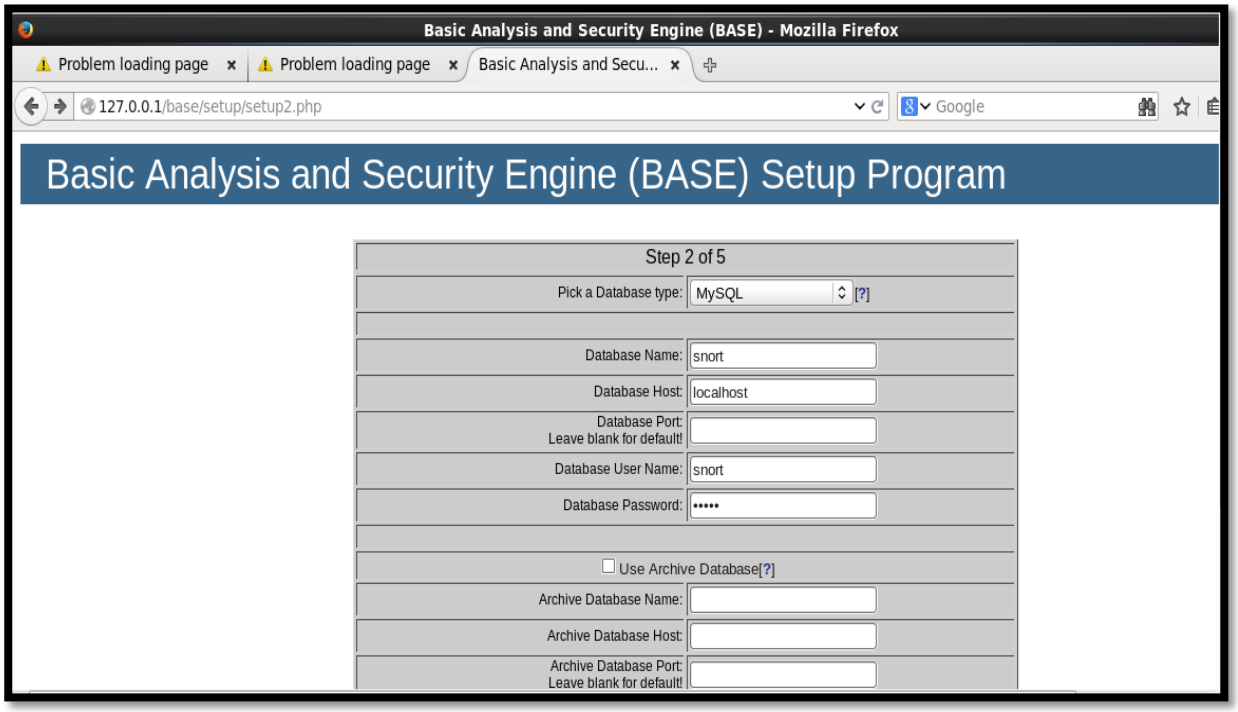

Figure IV.25 : Insertion des informations relatives à la base de données.

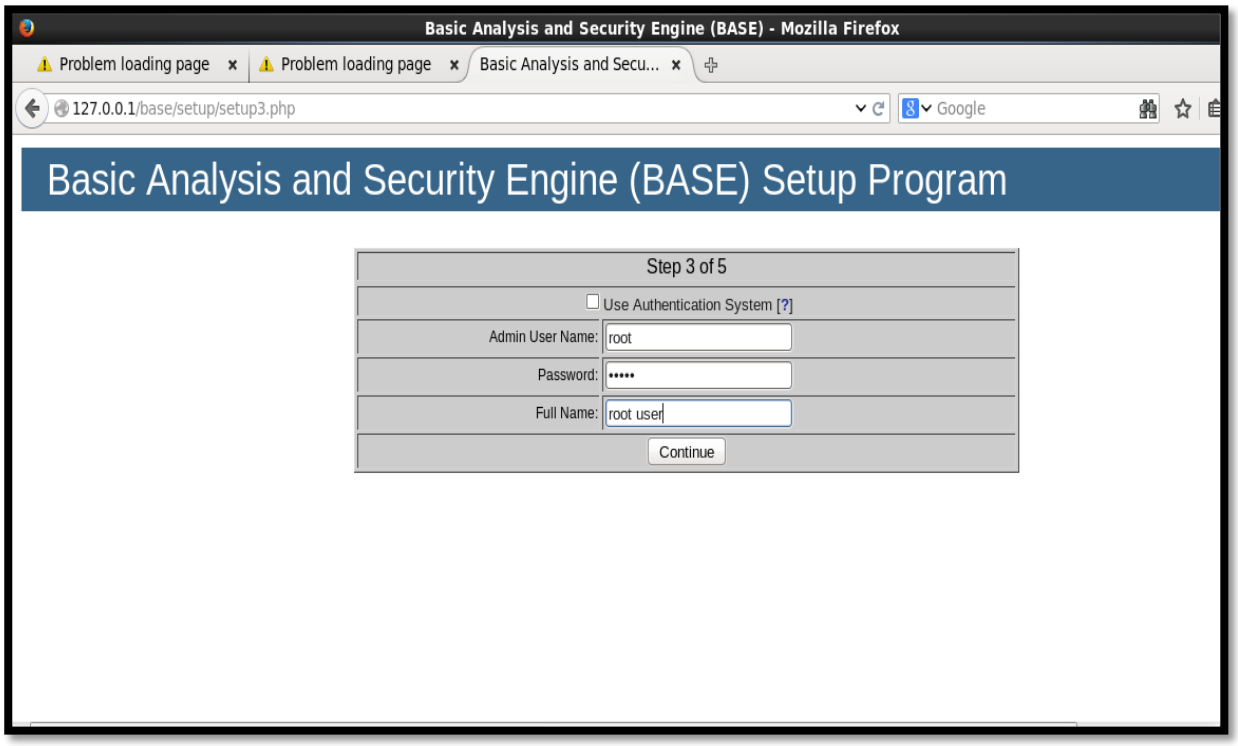

Figure IV.26 : Insertion des coordonnées de l'administrateur.

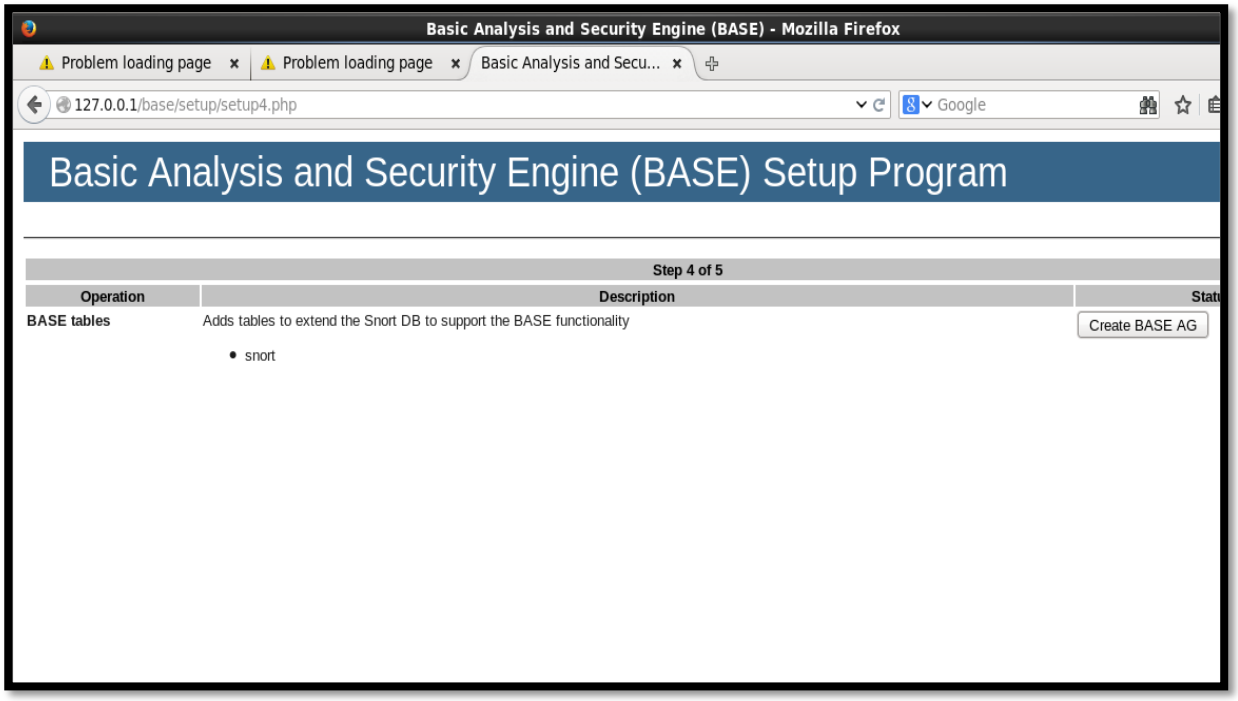

Figure IV.27 : Ajout des extensions de B.A.S.E.

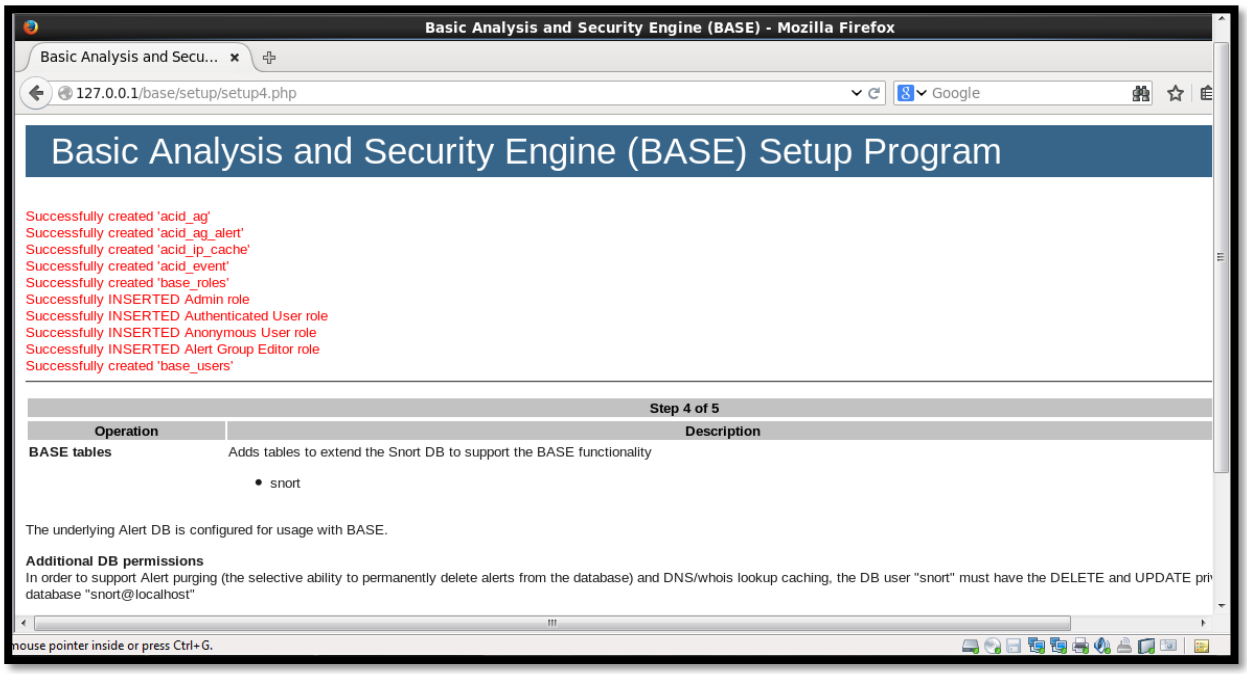

Figure IV.28 : Résultat de l'installation de B.A.S.E.

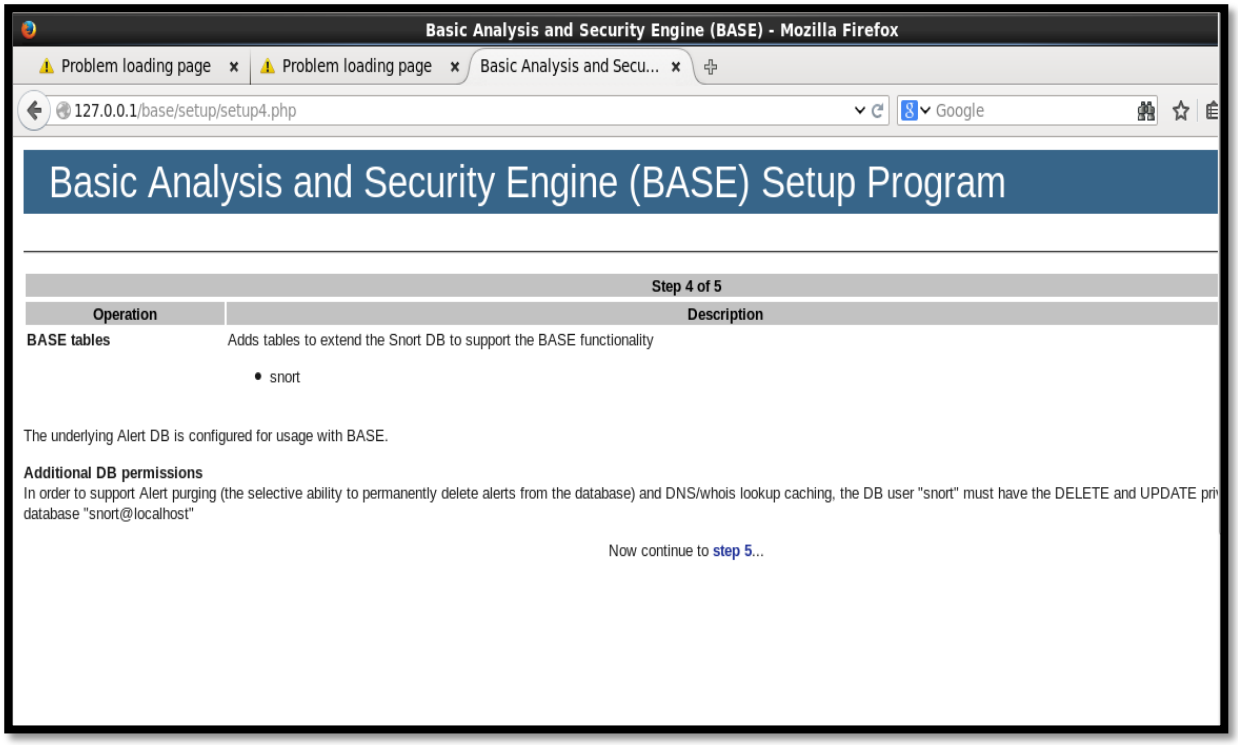

Figure IV.29 : Suite de la page précédente

IV.10. Lancement d'attaque

Scenario 1 : Attaque avec la méthode TCP.

Afin d'y parvenir à réaliser le scénario 1, on spécifie l'adresse IP cible et on sélectionne la méthode TCP dans l'outil d'attaque. Ensuite, on lace l'attaque.

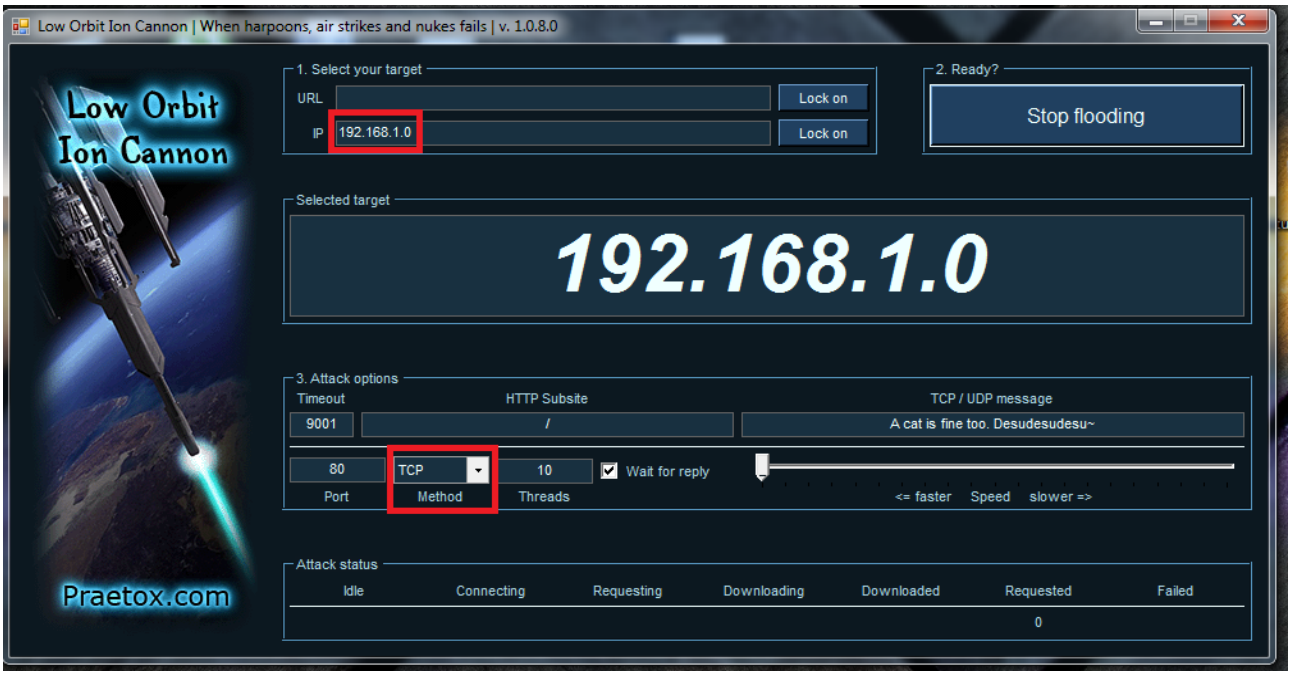

Figure IV.30 : Lancement d'une attaque sur le port TCP.

Le résultat du scénario précédent est affiché au niveau de la console B.A.S.E comme suit :

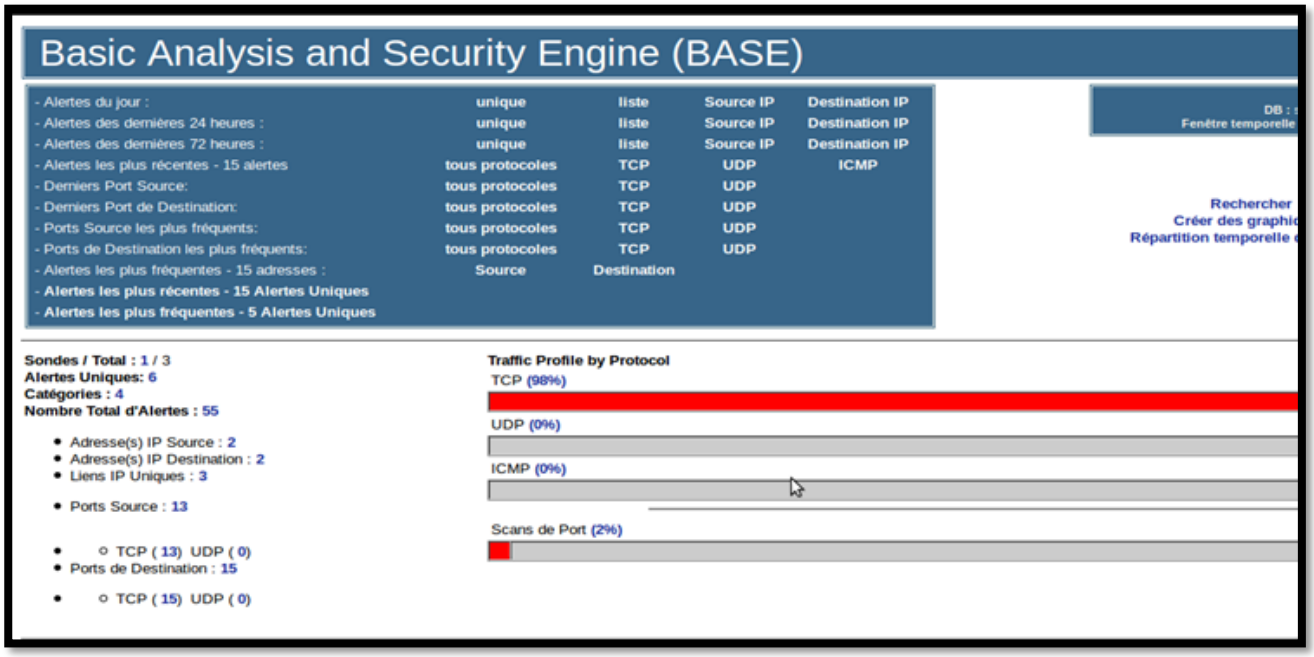

Figure IV.31 : Détection d'une attaque sur le port TCP.

Scénario 2 : Attaque avec la méthode UDP

La seule différence par rapport au scénario 1 réside dans la méthode sélectionnée, dans ce cas on choisit UDP.

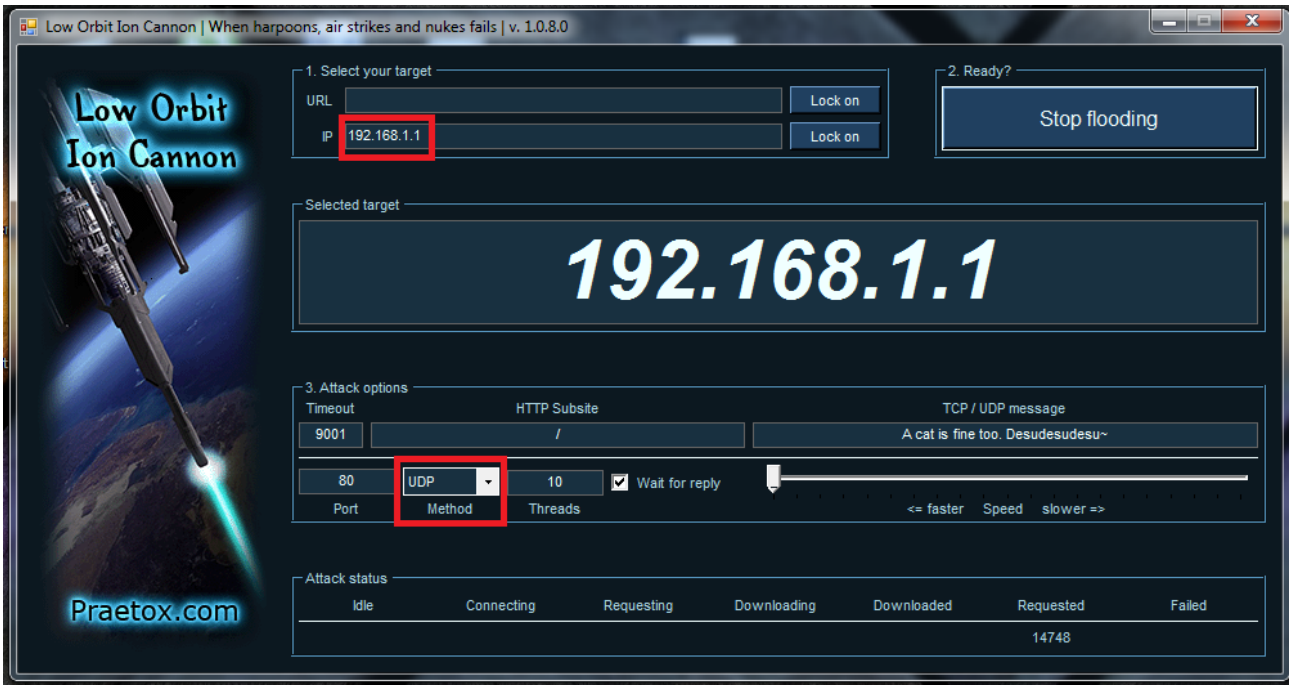

Figure IV.32 : Lancement d'une attaque sur le port UDP.

Le résultat de ce scénario est le suivant :

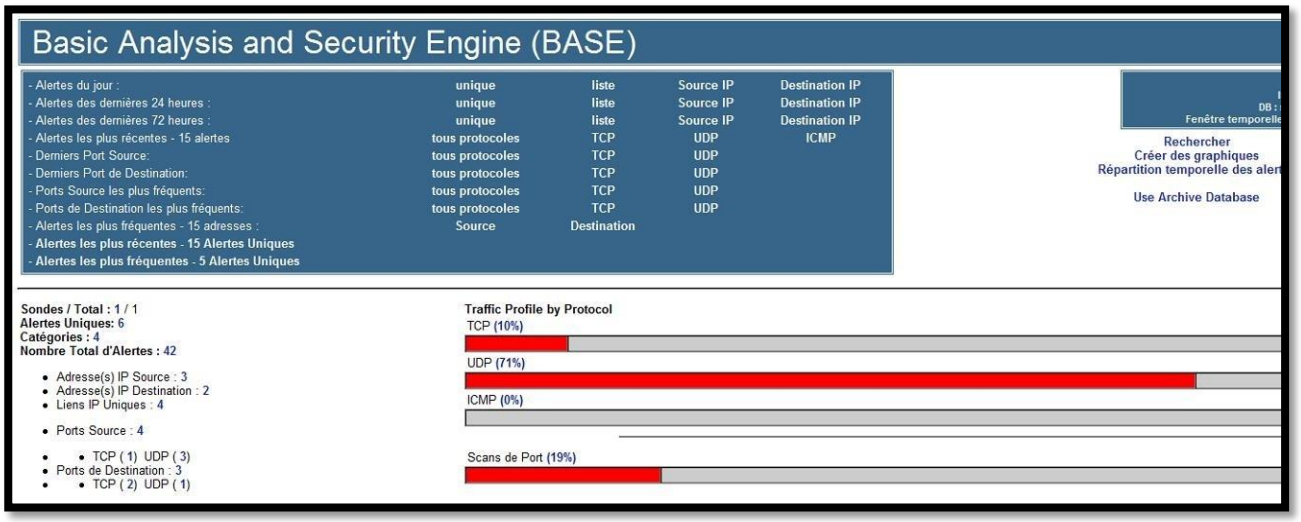

Figure IV.33 : Détection d'une attaque sur le port UDP.

Conclusion :

Dans cette partie, nous avons illustré l'installation et le mécanisme de fonctionnement de Snort en détails. Nous avons vu à la fin, comment Snort a pu stopper une attaque DoS avec succès.

Snort est un outil Open source qui est surtout conseillé aux petites entreprises qui n'ont pas les moyens de se procurer des solutions hardwares qui sont très chères et qui offrent le moindre service ou mise à jour avec des tarifs exorbitants.

Conclusion générale

L'évolution des réseaux informatiques, notamment Internet, au cours de ces dernières années a engendré une augmentation considérable du nombre d'utilisateurs. Ceci est dû à essentiellement à la facilité d'accès et la diversité des services utiles offerts.

L'ouverture de ces réseaux rend l'accès aux informations plus simple et plus rapide, et les rend plus vulnérables et plus exposés aux menaces. Ainsi, la mise en place d'une politique de sécurité permettant de garantir la protection de ces réseaux des risques les plus courants est plus qu'indispensable.

Sur Internet, les pirates emploient de plus en plus de nouvelles stratégies pour dissimuler leurs caractères intrusifs afin y d'atteindre leurs objectif sans se faire détecter par les systèmes de sécurité mis en place. D'où la nécessité de mettre en place toute une politique de sécurité pour se prévenir contre ces attaques. Les systèmes de détection d'intrusions ne représentent qu'un maillon de cette politique, qui reste toute fois vulnérables eux aussi face aux attaques externes.

Nous avons abordé dans ce mémoire la détection d'intrusions dans les réseaux locaux, qui restent à présent un maillot indispensable dans la procédure de sécurisation d'un réseau informatique, en lui associant d'autres outils de sécurisation tels que les par-feux, les proxys, etc. Le but principal de ce projet était de mettre en place un système de détection d'intrusion afin de renforcer la sécurité d'un système d'information, à savoir de l'IDS Snort. Nous avons présenté l'aspect théorique concernant la sécurité des réseaux informatiques et les systèmes de détection d'intrusion, puis nous avons présenté quelques IDS existants dans la littérature, enfin et on a terminé notre travail par la mise en place de l'IDS Snort à savoir l'illustration des étapes de son installation et des tests d'évaluation réalisés.

Enfin, en conclusion, on va dire que ce mémoire nous a permis d'acquérir une certaine maitrise et un certain bagage dans le domaine de la sécurité informatique et plus particulièrement dans les IDS. Il faut savoir que les grandes menaces proviennent généralement de l'intérieur de l'entreprise, et non de l'extérieur. Des mots de passe simples, des droits d'accès trop élevés, des services mal configurés, ou encore des failles dans les logiciels restent la bête noire en matière de sécurité.

Bibliographie

[1] **P. BIONDI**. Architecture expérimentale pour la détection d'intrusions dans un système informatique. Avril-Septembre 2001.

[2] **A. ESSAIDI, V. BOISTUAUD et N. DIOP.** Conception d'une zone démilitarisée (DMZ). Mémoire de master en informatique, option réseau. Université de Marne la vallée. 2006-2007.

[3] **L. POINSOT.** Introduction à la sécurité informatique. Support de cours. Université Paris 13.

[4] Le grand livre de la sécurité informatique. SecuriteInfo. Edition du 6 novembre 1006

[5] **L. YVES**. La sécurité informatique, 2002.

[6] G. **STEPHANE.** Les types d'attaques. 2013.

[7] **G. PUJOLLE**. Les réseaux. Edition Eyrolles. 2008.

[8] **L. BLOCH, C. WOLPHUGEL.** La Sécurité informatique. Edition Eyrolles. 2007.

[9] **J. F. PILLOU, J. P. BAY**, Tout sur la Sécurité Informatique, Dunod, 4eme édition, 2016.

[10] **Y. DUCHEMIN**. Apporter les notions essentielles pour l'interconnexion de réseau dans des environnements de communication hétérogène basé sur TCP/IP. Avril 2000.

[11] **I. LABED**. Proposition d'un système immunitaire artificiel pour la détection d'intrusions. Mémoire de magister en informatique. Uuniversité de Constantine. 2006.

[12] **Z. ABDELHALIM**. Recherche et détection des patterns d'attaques dan les réseaux IP à haut débit. Thèse de doctorat. Université d'Evry Val d'Essonne. Janvier 2011.

[13] **M. AMAND et M. NSIRI**, Etude d'un système de détection d'intrusion comportemental pour l'analyse du trafic aéroportuaire, Rapport de projet tutoyé, janvier 2011.

[14] **Y. FARHAOUI**. Evaluation des Systèmes de Détection et de Prévention des Intrusions et la Conception d'un BiIDS, thèse doctorale, UNIVERSITE IBN ZOHR CENTRE DES ETUDES DOCTORALES IBN ZOHR, Maroc, décembre 2012.

[15] **C. MICHEL**. Langage de description d'attaque pour la détection d'intrusion par corrélation d'évènements ou d'alertes en environnement réseau hétérogène. Thèse de doctorat. Université de Rennes 1. Décembre 2003.

[16] **T. Evangelista.** Les systèmes de détection d'intrusion informatiques. édition DUNDO. Paris. 2004

[17] **N. BAUDOIN, M. KARLE.** NT Réseau IDS et IPS. Ingénieurs2000. 2003 / 2004.

[18] **S. EE. SMAHA.** Haystack : An Intrusion Detection System. In FourthAerospace Computer Security Applications Conference, pages 37-44. Austin, TX, 1988.

[19] **M. M. Sebring, E. Shellhouse, M. E. Hanna, and R. A. Whitehurst**. Expert system in intrusion detection : A case study. In Proceedings of the 11thNational Computer Security Conference, pages 74–81. 1988.

[20] T.**F.Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, P. G. Neumann, H.S. Javitz, and A. Valdes**. Ides : Theenhanced prototype, a real-time intrusion detection system. Technical Report SRI Project 4185–010, SRI-CSL-88–12, CSL SRI International, Computer Science Laboratory, SRI Intl.October 1998.

[21] **S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle.** Griids - a graph basedintrusion detection system for large networks, 1996.

Résumé

Les systèmes d'information sont, aujourd'hui, de plus en plus ouverts sur le monde extérieur notamment Internet. Cette ouverture simplifie considérablement la vie pour l'homme en lui offrant plusieurs services, et ce à travers des centaines de millions d'ordinateurs reliés à internet. Cependant, cette interconnexion desordinateurs permet également aux utilisateurs malveillants d'utiliser ces ressources à des fins abusives et de lancer des attaques informatiques de divers types à l'encontre desserveurs web.

La sécurité des systèmes informatiques est une problématique d'une importance capitale pour les individus ainsi que pour les entreprises. Elle repose en premier lieu sur la mise en place de toute une politique de sécurité autour de ces systèmes. Outre la mise en place de pare-feu,des systèmes d'authentification de plus en plus sécurisés, etc. Il est nécessaire, pour compléter cette politique de sécurité, d'avoir des outils de surveillance pour auditer les systèmes d'information et détecter d'éventuelles intrusions. Les systèmes de détection d'intrusions (intrusion détection system, IDS) ont été conçus pour surveiller les systèmes d'informations(SI) et découvrir les violations de la politique de sécurité automatiquement. C'est dans cette optique que s'inscrit notre travail, à savoir l'étude des systèmes d'intrusion dans les réseaux locaux et leur mise en œuvre.

Mots clés : attaques informatiques, sécurité des systèmes informatiques, IDS.

Abstract

Information systems are, today, more and more open to the outside world including the Internet. This openness greatly simplifies life for man by offering several services, and this through hundreds of millions of computers connected to the internet. However, this interconnection of computers also allows malicious users to use these resources for abusive purposes and to launch various types of computer attacks against web servers.

Security in computer systems is a vital problem for the individual as well as for the companies. It is based first and foremost on the implementation of a whole security policy around these systems. In addition to setting up firewalls, incre a singly secure authentication systems, etc. It is necessary, in addition to this security policy, to have monitoring tools to audit information systems and to detect possible intrusions. Intrusion detection systems (IDS) were designed to monitor information systems (SI) and to automatically discover security policy violations.It is in this perspective that our work is part, which consists of the study of the intrusion systems in the local networks and their implementation.s

Keywords : computer attack, Security in computer systems, IDS.