**REPUBLIQUE ALGERIENNE DEMOCRATIQUE 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 Master en Informatique**

**Option**
**Administration et Sécurité des Réseaux**

**Thème**

# Etude et mise en place d'un système de Détection d'intrusion sous Linux

| | | |
|---|---|---|
| **Présenté par :** | **Mr Farah Zoubeyr** | **encadreur** |
| **Saci** Souhila | **Mr Atrouche Abdelghani** | **Co-encadreur** |
| **Batouche** Sonia | **Mr Akilal abdellah** | **Examinateur** |
| | **Mr Mehoued Kamel** | **Président** |

**Promotion 2014/2015**

# Table des matières

# Liste des abréviations

| | |
|---|---|
| **IDS** | **I**ntrusion **D**etection **S**ysteme |
| **IPS** | **I**ntrusion **P**révention **S**ystème |
| **NIDS** | **N**etwork **I**ntrusion **D**etection **S**ysteme |
| **NIPS** | **N**etwork **I**ntrusion **P**révention **S**ystème |
| **KIPS** | **k**ernel Intrusion **P**révention **S**ystème |
| **HIPS** | **H**ost **I**ntrusion **P**révention **S**ystème |
| **TCP** | **T**ransmission **C**ontrol **P**rotocol |
| **IP** | **I**nternet **P**rotolcol |
| **HTTP** | **H**ypertext **T**ransfer **P**rotocol |
| **UDP** | **U**ser **D**atagram **P**rotocol |
| **DMZ** | **D**emilitarized **Z**one |
| **B.A.S.E** | **B**asic **A**nalysis and **S**ecurity **E**ngine |
| **DOD** | **D**epartment of **D**efance |
| **DHCP** | **D**ynamic **H**ost **C**onfiguration **P**rotocol |
| **LAN** | **L**ocal **A**rea **N**etwork |
| **WAN** | **W**ide Area **N**etwork |
| **OSI** | **O**pen **S**ysteme **I**nterconnexion |
| **LOIC** | **L**ow **O**rbit **I**on **C**annon |

# Liste des figures

# Liste des tableaux

# Introduction générale

Devant la complexité croissante des réseaux qui est devenu de plus en plus gigantesque et étendue dans le domaine professionnel ainsi que pour les particuliers, on se trouvera devant le défi de se contribuer à la recherche des solutions pour se protéger contre les pirates et les malware qui sont de plus en plus nombreux et diversifiées les un que les autres grâce au réseau internet.

Les anti-virus et les pare-feux représentent des solutions de protection mais qui s'avèrent limitées face au développement rapide des techniques de piratage, d'où la nécessité de mettre en place un système de détection et de prévention d'intrusion(IDS), qui est une méthode de surveillance permanente du système afin de détecter toute violation de la politique de sécurité et signaler les attaques portant atteinte à la sécurité du réseau informatique.

L'objectif de notre travail est de mettre en place Snort, qui est un système de détection d'intrusion réseau (NIDS) open source, disponible sous licence GPL, fonctionnant sur les systèmes Windows et linux.il est capable d'effectuer en temps réel des analyses de trafic et de loger les paquets sur un réseau IP. [19]

Notre mémoire est organisé comme suit :

La première partie consiste à présenter l'organisme d'accueil qui nous a pris en charge durant toute la période de notre stage pratique.

La seconde partie de notre travail est consacré à la présentation des différents aspects de la sécurité informatique ainsi que les concepts relatifs aux réseaux d'une manière générale.

La troisième partie comprend une description bien détaillée des systèmes de détection et de prévention d'intrusion (leurs différents types, leurs principes de fonctionnement, une comparaison entre IDS et IPS et leurs inconvénients).

La quatrième partie est réservée à l'illustration de l'outil IDS qui est Snort que nous avons proposé comme solution ainsi que son mode de fonctionnement, son architecture et sa position dans le réseau.

La dernière partie consiste à détailler les modules et dépendances à prendre en compte dans l'installation de SNORT sous CentOs, ainsi que toutes les configurations nécessaires afin de le rendre fonctionnel.

## Introduction

Le groupe Cevital c'est constitué au fil des investissements, autour de l'idée forte de bâtir un ensemble économique. Porté par plus de 10200 collaborateurs, il représente le fleuron de l'économie algérienne. Le fondateur du groupe Cevital résume les clefs du succès en sept points : le réinvestissement systématique des gains dans des secteurs porteurs à forte valeur ajoutée, la recherche et la mise en œuvre des savoir-faire technologiques les plus évolués, l'attention accordée aux choix des hommes et des femmes, à leurs formations et aux transfert des compétences, l'esprit d'entreprise, le sens de l'innovation, la recherche de l'excellence et la fierté et la passion de servir l'économie nationale.

## I .1. Présentation du complexe Cevital

Cevital est un groupe familial de vingt-cinq sociétés, réparties dans cinq secteurs d'activités : L'Industrie Métallurgique, l'Information et la Communication, la Distribution Automobile, le Transport Terrestre et Maritime, l'Industrie Agroalimentaire. CEVITAL est parmi les entreprises qui ont vu le jour dés l'entrée du pays dans l'économie de marché. Disposant de technologies de pointe.

Cevital possède deux raffineries : une d'huile et l'autre de sucre.

La raffinerie d'huile alimentaire a été mise en chantier en Mai 1998, en Aout 1999 elle est rentré en production, plus tard en 2000, la raffinerie du sucre est mise en chantier, elle n'est devenue fonctionnel qu'en 2002.

Un autre produit est mis en chantier en 2000 et en production en 2001, c'est la margarine.

Une deuxième raffinerie de sucre de 3000 T, de plus le silo sucre blanc 80000 T et le silo sucre roux 150000 T, une unité d'eau minéral L'alla Khadîdja, et une autre unité de Cojek a El Kseur. Enfin, une station de cogénération.

### I.1.1. Organigramme de groupe Cevital

Voici le schéma général du groupe Cevital, dont chaque direction a pour but d'assurer le bon fonctionnement de chaque partie du groupe comme le montre cette figure :

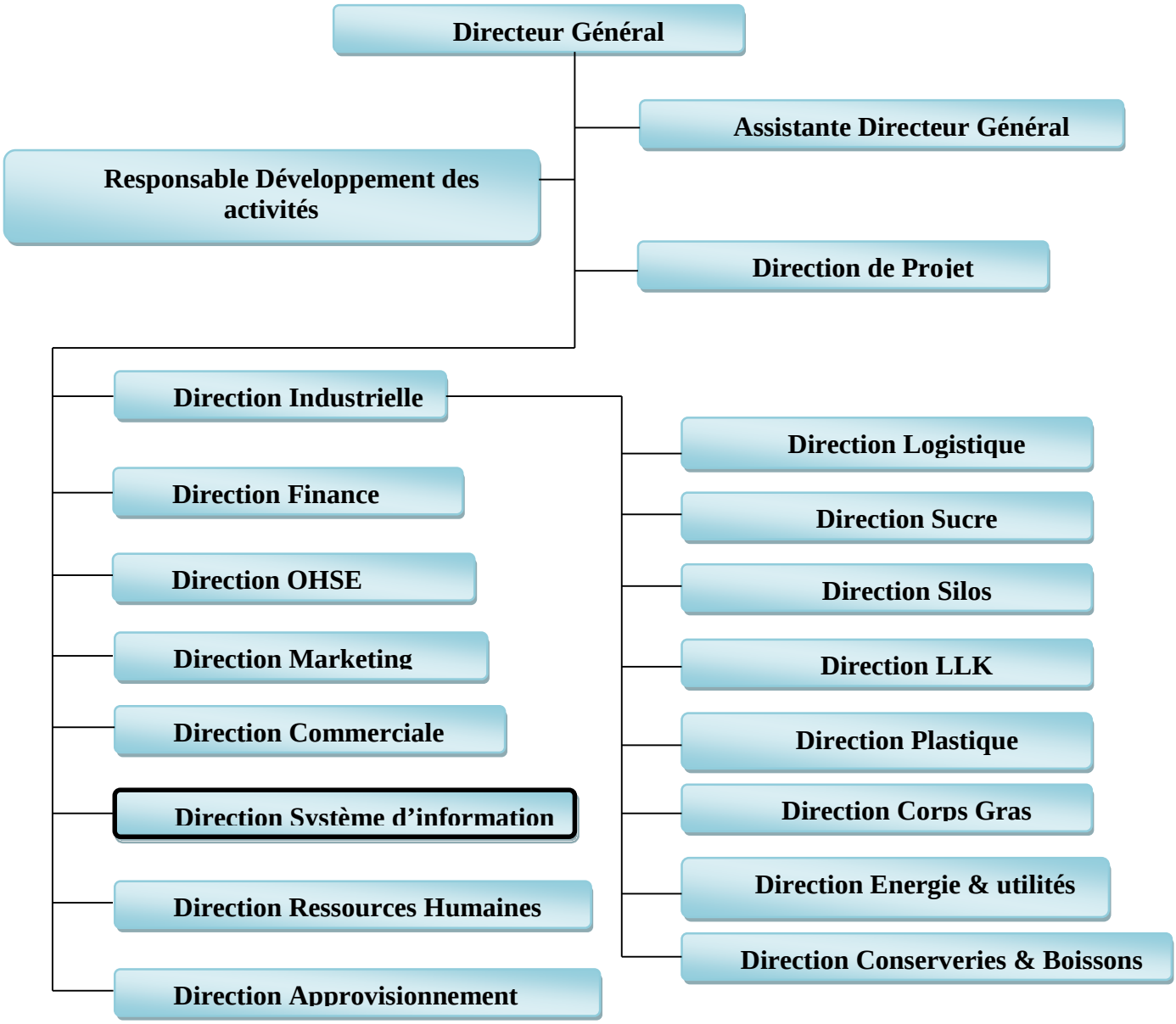



**Figure 01** : Organigramme du groupe Cevital

❖ **Les missions**

L'entreprise a pour missions principales de développer la production et d'assurer la qualité du conditionnement des huiles, des margarines et du sucre à des prix nettement plus compétitifs et cela dans le but de satisfaire le client et de le fidéliser.

❖ **Les activités**

Lancé en Mai 1998, le complexe Cevital a débuté son activité par le conditionnement en Décembre 1998, en Fevrier 1991, les travaux de génie civil de la raffinerie ont débuté. Cette dernière est devenue fonctionnelle, en Août 1999. L'ensemble des activités de Cevital est concentré sur la production et la commercialisation des huiles végétales, de margarine et de sucre se présente comme suite :

-Raffinage d'huile 1600 T/J pouvant passer après extension à 1800 T/J.

- Production de margarine de capacité 600 T/J.

- Fabrication d'emballage en PET (9600 unités/h).

-Stockage céréales.

-Electrolyseur (par mesure de sécurité doit être déplacé hors Cevital).

-Extension de la sucrerie.

-Savonnerie.

-Minoterie.

-Hydroélectrique d'huile.

❖ **Les objectifs**

Les objectifs visés par Cevital peuvent se présenter comme suit :

- Encouragement des agricultures par des aides financières pour la production locale de graines oléagineuses.
- Importation de graines oléagineuses pour l'extraction directe des huiles brutes.
- Diversification de ses produits et sa diffusion sur tout le territoire national.
- Modernisation de ses installations et adoption de nouvelles démarches de gestion technique afin d'augmenter le volume de sa production.
- Positionner ses produits sur le marché étranger par leurs exportations.
- Optimisation de ses offres d'emploi sur le marché du travail.

## I.2. L'informatique dans Cevital

Cevital est parmi les entreprises possédant une direction informatique et donne une grande importance au domaine de l'informatique.

### I.2.1 Présentation de l'organisme d'accueil

Notre étude se focalise au niveau du groupe Cevital de Béjaïa ou nous avons effectué notre stage, dans la direction de l'informatique réseau et télécom.

❖ **Organigramme de la direction système d'information**

La direction système d'information de Cevital est composée de deux départements :

- ⇨ Métiers.
- ⇨ Département système réseaux télécom : il assure de bon fonctionnement de réseaux (internet) et même la télécommunication (téléphonie).

Chaque département a pour objectif d'améliorer le niveau de l'informatique et ces services pour garantir le développement et la progression des services du groupe Cevital.
L'organigramme de la direction système d'information est montré dans la figure suivante :

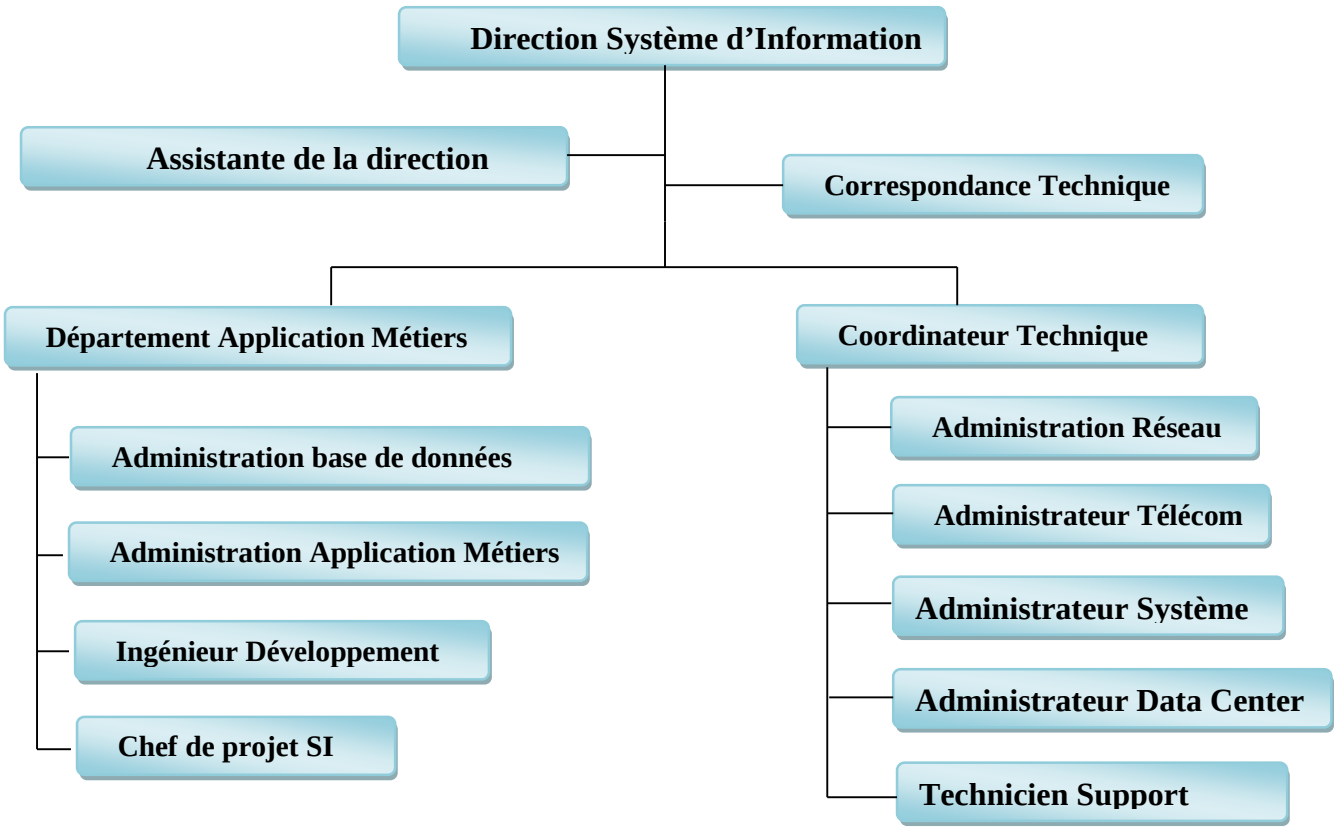



**Figure 02 :** Organigramme de la direction Système d'Information

### I.2.2. La structure du réseau de CEVITAL-BEJAIA

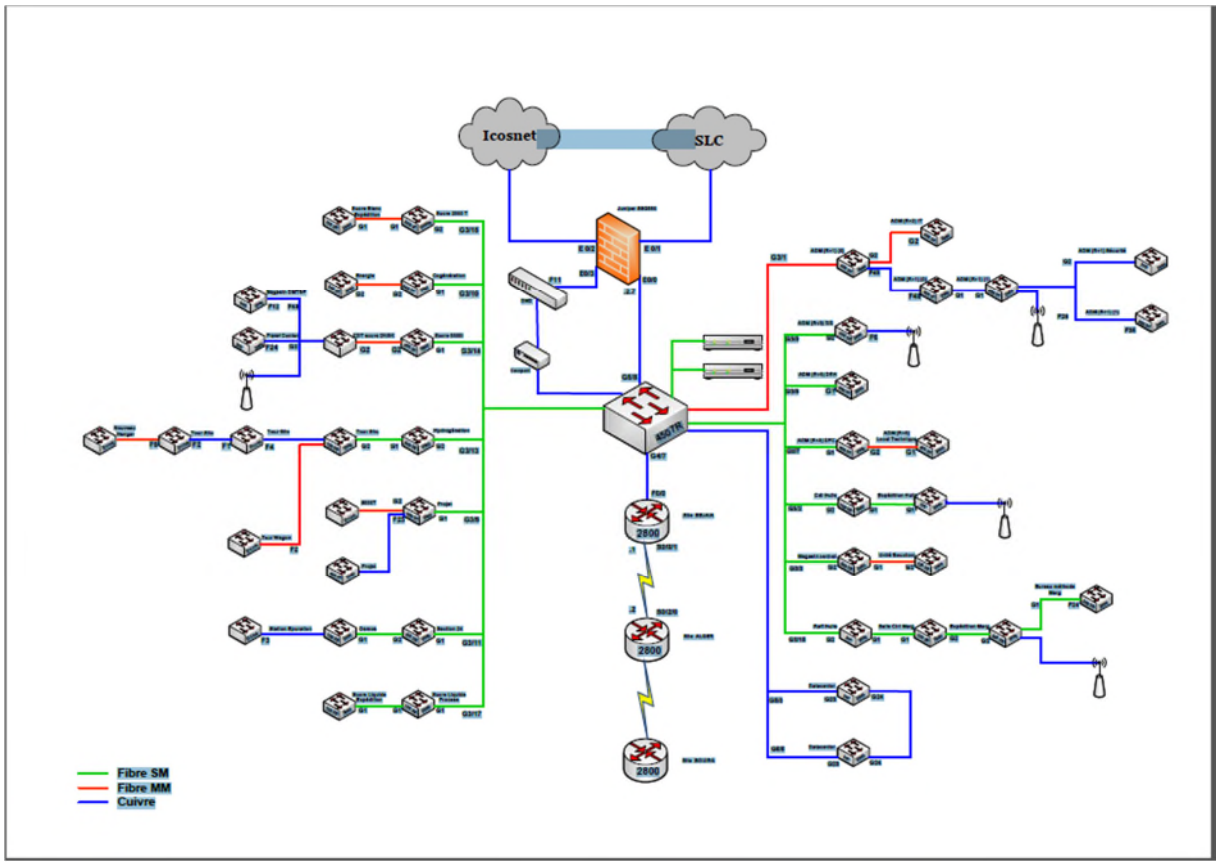



**Figure 03 :** Interconnexion Switch et points d'accès CEVITAL-BEJAIA

## I.3. Problématique

Les vulnérabilités et les failles des systèmes informatiques sont continuellement découvertes, et les risques d'attaques à distance ne font également qu'augmenter. Actuellement, les pare-feu permettent de réduire partiellement ces risques. Cependant un réseau protégé par un pare-feu demeure tout de même pénétrable vue qu'une menace peut accéder tout de même au réseau à travers ce dernier, ce qui est un problème important à gérer.

Si l'IDS est intégré dans un pare-feu, ces derniers peuvent entrer en conflits car ils analysent le même flux de données, ce qui provoque un plantage et un ralentissement considérable des performances de l'équipement en présence d'un grand trafic.

De ce fait, durant notre stage au sein de CEVITAL nous avons constaté des anomalies relatives à la sécurité de leur réseau, à savoir le manque d'un mécanisme de détection d'intrusion.

## I.4. Objectifs

La configuration d'un IDS augmente le degré de sécurité puisqu'il fonctionne indépendamment des autres équipements, ce qui lui permet une capacité de détection plus performante. Il joue le rôle d'un complément aux pare-feu en lui permettant une analyse plus intelligente du trafic.

Pour remédier au problème traité dans la problématique nous avons choisi de mettre en place un IDS (SNORT) sous le système d'exploitation LINUX qui sera placé entre le routeur et le pare-feu pour éviter les conflits. Voici les objectifs tracés :

- Etude du réseau existant CEVITAL et identification des besoins.
- Installation et mise en place d'un IDS.
- Configuration de l'IDS.

## Conclusion

On a consacrer ce chapitre a la présentation de l'organisme d'accueil et plus précisément le cadre de travail ou on a effectuer notre stage qui nous a permit de mieux comprendre et apprécier le travail abattu par l'ensemble du complexe CEVITAL , de comprendre la place qu'occupe cette structure dans le domaine, ainsi, l'étude du réseau CEVITAL nous a permis de bien comprendre son architecture et les stratégies utilisées pour sa sécurisation.

# Introduction

Les systèmes d'information sont aujourd'hui de plus en plus ouverts sur Internet. Cette ouverture, a priori bénéfique, pose néanmoins un problème majeur : il en découle un nombre croissant d'attaques.
La mise en place d'une politique de sécurité autour de ces systèmes est donc primordiale. Au cours de ce chapitre, nous verrons comment se protéger efficacement face à ces intrusions, mais aussi les problèmes techniques déduits de ces outils.
Mais avant cela, il est important, pour comprendre le rôle précis de ces systèmes, de faire un rappel des principales attaques existantes à l'heure actuelle.

## II.1. La sécurité informatique

### II.1.1. Qu'est ce que la sécurité ?

La sécurité informatique c'est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles. [13]

### II.1.2. Mise en place d'une politique de sécurité

La mise en place d'une politique de sécurité globale est assez difficile, essentiellement par la diversité des aspects à considérer. Une politique de sécurité peut se définir par un certain nombre de caractéristiques : les niveaux ou elle intervient, les objectifs de cette politique et enfin les outils utilisés pour assurer cette sécurité. Néanmoins, elle ne peut garantir une sécurité absolue.

#### II.1.2.1. Etude des risques

Il est nécessaire de réaliser une analyse de risque en prenant soin d'identifier les problèmes potentiels avec les solutions avec les coûts associés. L'ensemble des solutions retenues doit être organisé sous forme d'une politique de sécurité cohérente, fonction du niveau de tolérance au risque. Il faut cependant prendre conscience que les principaux risques restent : « câble arraché », « coupure secteur », « crash disque », « mauvais profil utilisateur »….
Voici quelques éléments pouvant servir de base à une étude de risque:

- Quelle est la valeur des équipements, des logiciels et surtout des informations ?
- Quel est le coût et le délai de remplacement ?
- Faire une analyse de vulnérabilité des informations contenues sur les ordinateurs en réseau (programmes d'analyse des paquets, logs…).
- Quel serait l'impact sur la clientèle d'une information publique concernant des intrusions sur les ordinateurs de la société ?

#### II.1.2.2. Différents aspects de la sécurité

Une politique de sécurité s'élabore à plusieurs niveaux :

- Il faut tout d'abord sécurisé l'accès aux données de façon logicielle (authentification, contrôle d'intégrité).
- Il faut également sécuriser l'accès physique aux données : serveurs placés dans des salles blindées (qui empêchent les ondes électromagnétiques d'être captées) avec badge d'accès…
- Un aspect très important pour assurer la sécurité des données d'une entreprise est de sensibiliser les utilisateurs aux notions de sécurité, de façon à limiter les comportements à risque.

De même, si les utilisateurs laissent leur mot de passe écrit à côté de leur ordinateur , son utilité est limitée…

- Enfin, il est essentiel pour un responsable de sécurité de s'informer continuellement, des nouvelles attaques existantes, des outils disponibles… de façon à pouvoir maintenir à jour son système de sécurité et à combler les brèches de sécurité qui pourraient exister. [2]

#### II.1.2.3. Services de la sécurité (Objectifs)

Les systèmes d'information représentent l'ensemble des données de l'entreprise et les infrastructures matérielles et logicielles. La sécurité informatique d'une manière générale, consiste à assurer que les ressources d'une organisation, soient uniquement utilisées dans le cadre prévu.

a) **La confidentialité :** les données ne doivent être visibles que pour les personnes autorisées.
b) **L'authentification :** consiste à assurer l'identité d'un utilisateur, c'est-à-dire garantir à chacun des correspondants, que son partenaire est bien celui qu'il croit être.
c) **L'intégrité :** il faut pouvoir garantir que les données protégées n'ont pas été modifiées par une personne non autorisée. Le but étant de ne pas altérer les informations sensibles de l'entreprise.
d) **La non-répudiation :** il s'agit de garantir qu'aucun des correspondants ne pourra nier la transaction effectuée.
e) **La disponibilité :** les données doivent restées accessibles aux utilisateurs. C'est la capacité à délivrer un service permanent à l'entreprise.

#### II.1.2.4. Principaux défauts de la sécurité informatique

Les défauts de sécurité d'un système d'information les plus souvent constatés sont :

- Installation des logiciels et matériels par défaut.
- Mises à jour non effectuées.
- Mots de passe inexistants ou par défaut.
- Services inutiles conservés (Netbios…).
- Traces inexploitées.
- Pas de séparation des flux opérationnels des flux d'administration des systèmes.
- Télémaintenance sans contrôle fort.
- Procédures de sécurité obsolètes (périmés).
- Authentification faible.

### II.1.3. attaques informatiques

Tout ordinateur connecté à un réseau informatique, est potentiellement vulnérable à une attaque

### II.1.3.1. Anatomie d'une attaque

Fréquemment appelés « les 5 P » dans la littérature, ces cinq verbes anglophones constituent le squelette de toute attaque informatique : Probe, Penetrate, Persist, Propagate, Paralyze.

- **Probe** (Analyser) : Dans un premier temps, une personne mal intentionnée va chercher les failles pour pénétrer le réseau.
- **Penetrate** (Pénétrer) : Une fois une ou plusieurs failles identifiées, le pirate va chercher à les exploiter afin de pénétrer au sien du SI.
- **Persist** (Persister) : une fois le réseau infiltré, le pirate cherchera à y revenir facilement. Pour cela, il installera par exemple des back doors. Cependant, en général, il corrigera la faille par laquelle il s'est introduit afin de s'assurer qu'aucun autre pirate n'exploitera sa cible.
- **Propagate** (Propager) : Le réseau est infiltré, l'accès est facile. Le pirate pourra alors explorer le réseau et trouver de nouvelles cibles qui l'intéresseraient.
- **Paralyze** (Paralyser) : Les cibles identifiées, le pirate va agir et nuire au sein du SI.[3]

### II.1.3.2. Cyber-attaques

Longtemps, les organismes étatiques se sont cantonnés à un rôle de veille, d'alerte, de recueil et de renseignements. Aujourd'hui, un rôle défensif leur est officiellement assigné. Cela signifie qu'ils doivent coordonner l'action des services de l'État pour la mise en œuvre de leur cyber-défense. On peut cependant imaginer qu'en cas d'attaque contre des infrastructures vitales avec des conséquences humaines, un État pourrait considérer cela comme un acte de guerre, et agir en conséquence : c'est-à-dire riposter. Cependant aucun État n'a, pour le moment, révélé l'existence officielle d'un programme de cyber contre-attaque. [6]

### II.1.3.3. Différents types d'attaques

Il existe deux types d'attaques :

#### a) Attaques réseaux

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

- **Les techniques de scan :** Le but des scans est de déterminer quels sont les ports ouverts, et donc en déduire les services qui sont exécutés sur la machine cible. Il existe un nombre important de techniques de scan. Idéalement, la meilleure technique de scan est celle qui est la plus furtive afin de ne pas alerter les soupçons de la future victime. Voici une description des techniques de scan les plus répandues :
  - **le scan simple** : aussi appelé le scan connect(), il consiste à établir une connexion TCP complète sur une suite de ports. S'il arrive à se connecter, le port est ouvert ; sinon, il est fermé. Cette méthode de scan est très facilement détectable.
  - **le scan furtif** : aussi appelé scan SYN, il s'agit d'une amélioration du scan simple. Ce scan essaie également de se connecter sur des ports donnés, mais il n'établit pas complètement la connexion : pas de commande ACK (acquittement) après avoir reçu l'accord de se connecter. Grâce à ceci, la méthode est bien plus furtive que le scan normal.
  - **les scans XMAS, NULL et FIN** : se basent sur des détails de la RFC du protocole TCP pour déterminer si un port est fermé ou non en fonction de la réaction à certaines requêtes. Ces scans sont moins fiables que le scan SYN, mais ils sont un

peu plus furtifs. La différence entre ces trois types de scan se situe au niveau des flags TCP utilisés lors de la requête.

- **le scan à l'aveugle** : s'effectue via une machine intermédiaire et avec du spoofing. Le système attaqué pense que le scan est réalisé par la machine intermédiaire et non par le pirate.
- **le scan passif** : est la méthode la plus furtive. Consiste à analyser les champs d'en-tête des paquets (TTL, ToS, MSS…) et à les comparer avec une base de signatures qui pourra déterminer les applications qui ont envoyé ces paquets.

❖ **IP Spoofing** : Le spoofing consiste à se faire passer pour un autre système en falsifiant son adresse IP. Le pirate commence par choisir le système qu'il veut attaquer. Après avoir obtenu le maximum de détails sur se système cible, il détermine les systèmes ou adresses IP autorisés à se connecter au système cible. Le pirate ensuite attaque le serveur cible en utilisant l'adresse IP falsifier.

❖ **ARP Spoofing** : son but est de rediriger le trafic d'une machine vers une autre. Grâce à cette redirection, une personne mal intentionnée peut se faire passer pour une autre. De plus, le pirate peut router les paquets qu'il reçoit vers le véritable destinataire, ainsi l'utilisateur usurpé ne se rendra compte de rien.

❖ **DNS Spoofing** : Son but est de fournir de fausses réponses aux requêtes DNS, c'est-à-dire indiquer une fausse adresse IP pour un nom de domaine. C'est de rediriger, à leur insu, des internautes vers des sites pirates. Grâce à cette fausse redirection, l'utilisateur peut envoyer ses identifiants en toute confiance par exemple.

❖ **Fragments attacks :** le but de cette attaque est de passer outre les protections des équipements de filtrage IP. un pirate peut par exemple s'infiltrer dans un réseau pour effectuer des attaques ou récupérer des informations confidentielles.

❖ **TCP Hijacking :** le but de cette attaque est de rediriger un flux TCP afin de pouvoir outrepasser une protection par mot de passe. le contrôle d'authentification s'effectuant uniquement à l'ouverture de la session, un pirate réussissant cette attaque parvient à prendre possession de la connexion pendant toute la durée de la session.[3]

### b) Attaques applicatives

Les attaques applicatives se basent sur des failles dans les programmes utilisés, ou encore des erreurs de configuration.

❖ **Les problèmes de configuration :** Une mauvaise configuration d'un serveur peut entraîner l'accès à des fichiers importants, ou mettant en jeu l'intégrité du système d'exploitation. C'est pourquoi il est important de bien lire les documentations fournies par les développeurs afin de ne pas créer de failles.

❖ **Les bogues** : Liés à un problème dans le code source, ils peuvent amener à l'exploitation de failles.

❖ **Les buffers overflows :** Les buffers overflows, ou dépassement de la pile, sont une catégorie de bogue particulière. Issus d'une erreur de programmation, ils permettent l'exploitation d'un shellcode à distance. Ce shellcode permettra à une personne mal intentionnée d'exécuter des commandes sur le système distant, pouvant aller jusqu'à sa destruction.

❖ **Les scripts :** Principalement web, ils s'exécutent sur un serveur et renvoient un résultat au client. Cependant, lorsqu'ils sont dynamiques, des failles peuvent apparaître si les entrées ne sont pas correctement contrôlées.

❖ **Les injections SQL :** Tout comme les attaques de scripts, les injections SQL profitent de paramètres d'entrée non vérifiés. Comme leur nom l'indique, le but des injections SQL est d'injecter du code SQL dans une requête de base de données. Ainsi, il est possible de récupérer des informations se trouvant dans la base ou encore de détruire des données.

❖ **Man in the middle :** cette attaque permet de détourner le trafic entre deux stations. Imaginons un client C communiquant avec un serveur S. Un pirate peut détourner le trafic du client en faisant passer les requêtes de C vers S par sa machine P, puis transmettre les requêtes de P vers S. Et inversement pour les réponses de S vers C. Totalement transparente pour le client, la machine P joue le rôle de proxy. Elle accédera ainsi à toutes les communications et pourra en obtenir les informations sans que l'utilisateur s'en rende compte. [3]

### II.1.4. Démarches pour anticiper et résoudre les problèmes

La variété et la disponibilité des outils d'attaques augmentent le risque des intrusions. Par conséquent les administrateurs s'appuient sur diverses solutions dans le but de maintenir la protection du réseau informatique. Voici quelques solutions proposés :

#### II.1.4.1. Les firewalls

Un pare-feu (Firewall) est un système physique ou logique qui inspecte les paquets entrants et sortants du réseau afin d'autoriser ou d'interdire leur passage en se basant sur un ensemble de règles appelées ACL. Il enregistre également les tentatives d'intrusions dans un journal transmis aux administrateurs du réseau et permet de contrôler l'accès aux applications et d'empêcher le détournement d'usage. [8]

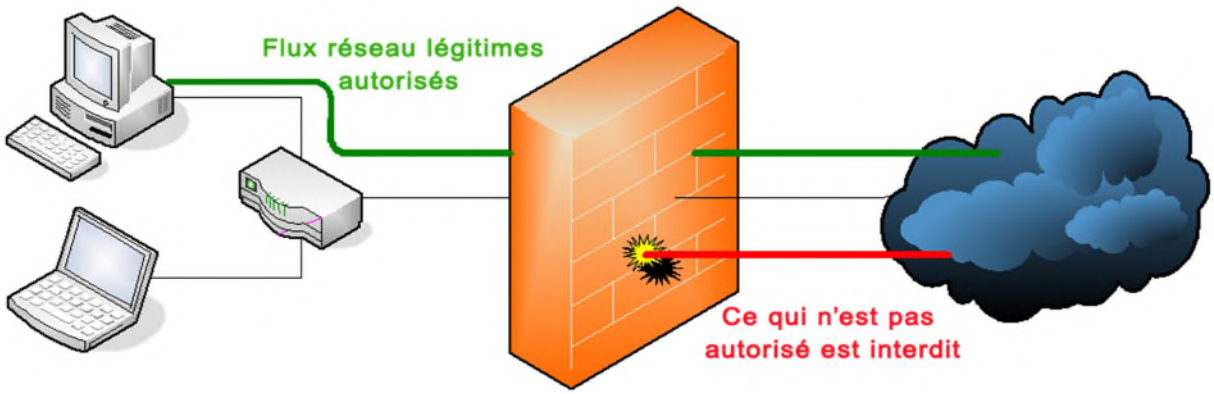



**Figure 4 :** l'emplacement d'un pare-feu dans un réseau

### II.1.4.2. Un serveur proxy (serveur mandataire)

Un proxy est un ensemble de processus permettant d'éliminer la connexion directe entre les applications des clients et les serveurs. Les organisations utilisent les proxys pour permettre à des machines de leur réseau d'utiliser Internet sans risque et sans que les utilisateurs externes ne soient capables d'accéder à ce réseau.

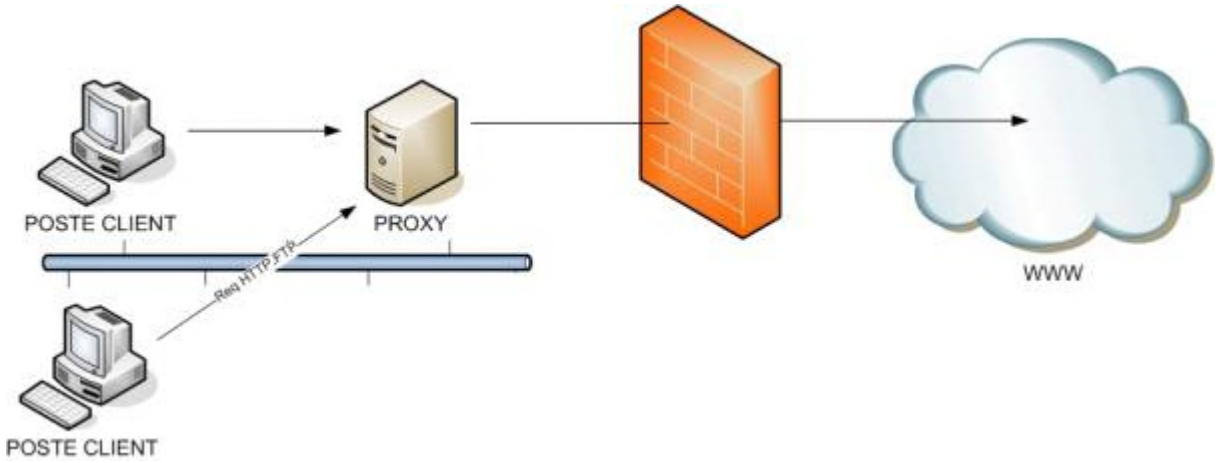



**Figure 5:** l'emplacement d'un proxy dans un réseau

## II.1.4.3. DMZ (Demilitarized zone)

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 devront 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. [4]

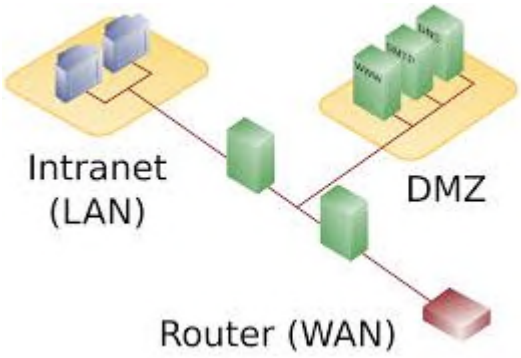



**Figure 6 :** La DMZ entre LAN et WAN

### II.1.4.4. Antivirus

Les antivirus sont des programmes qui permettent de détecter la présence de virus sur un ordinateur et de les supprimer.

L'éradication d'un virus est le terme utilisé pour le nettoyage d'un ordinateur. Il y a plusieurs méthodes d'éradication :

- Nettoyez le fichier infecté en supprimant le code malveillant.

- Retrait du fichier infecté entièrement.
- La mise en quarantaine le fichier infecté, qui consiste à le déplacer à un emplacement où il ne peut être exécuté. [11]

## II.2. Les réseaux

### II.2.1 La norme OSI

#### II.2.1.1 Définition

L'Open System Interconnection est une norme établie par Internationnal Standard Organisation, afin de permettre aux systèmes ouverts (ordinateur, terminal, réseau, …) d'échanger des informations avec d'autres équipements hétérogènes. Cette norme est constituée de 7 couches, dont 4 premières sont dites basses et les 3 supérieures dites hautes. Le principe est simple, la couche la plus basse (directement au dessus du support physique) ne peut communiquer directement avec une couche n+1 : chacune des couches est composée d'éléments matériels et/ou logiciels chargés de « transporter » le message à la couche supérieure. [16]

| |
|---|
| 7. Application |
| 6. Présentation |
| 5. Session |
| 4. Transport |
| 3. Réseau |
| 2. Liaison |
| 1. Physique |

**Figure 7 :** Le modèle OSI

#### II.2.1.2. Les différentes couches du modèle OSI

a) La couche physique s'occupe de la connexion physique sur le réseau. Elle se charge de la transmission de bits à l'état brut sur un canal de transmission.
b) La couche Liaison à pour but de transmettre les données sans erreur. Elle décompose les données de l'émetteur en trames de données puis les envoie de façon séquentielle. Différentes méthodes permettant de protéger les données contre les erreurs sont utilisées, comme le codage de détection et de correction d'erreur.
c) La couche Réseau assure la commutation et le routage des paquets entre les nœuds du réseau. Pour le modèle OSI, il existe deux méthodes principales d'acheminement: la commutation de circuits et la commutation de paquets. C'est cette couche qui gère les congestions sur les nœuds du réseau.
d) La couche Transport permet l'établissement, le maintien et la rupture des connexions. L'une des tâches principales de cette couche est d'accepter des

données de la couche supérieure et de les diviser en unités plus petites : il s'agit de l'opération de fragmentation. Elle offre un service réel de bout en bout de la source à la destination, indépendant du chemin effectif utilisé entre les machines.

e) La couche Session permet d'établir une connexion logique entre deux applications. Elle assure l'organisation et la synchronisation du dialogue.
f) La couche Présentation s'occupe de la syntaxe des données. Cette couche permet à deux machines de communiquer même lorsqu'elles utilisent des représentations de données différentes. Elle gère des structures de données haut niveau pour accomplir cette tâche.
g) La couche Application fournit les services et interfaces de communication aux utilisateurs. [1]

## II.2 Le TCP/IP

### II.2.1 Définition

Dans les années 1970, le département de la Défense américain, ou DOD (Department Of Defense), décide, devant le foisonnement de machines utilisant des protocoles de communication différents et incompatibles, de définir sa propre architecture. Cette architecture, dite TCP/IP, est à la source du réseau Internet. Elle est aussi adoptée par de nombreux réseaux privés, appelés intranet.
Les deux principaux protocoles définis dans cette architecture sont les suivants :
• IP (Internet Protocol), de niveau réseau, qui assure un service sans connexion.
• TCP (Transmission Control Protocol), de niveau transport, qui fournit un service fiable avec connexion. [5]

### II.2.2 Découpage en couche

Le protocole TCP/IP étant antérieur au modèle OSI, il ne respecte pas réellement celui-ci. Cependant, on peut faire correspondre les différents services utilisés et proposés par TCP/IP avec le modèle OSI, et obtenir ainsi un modèle en 4 couches.

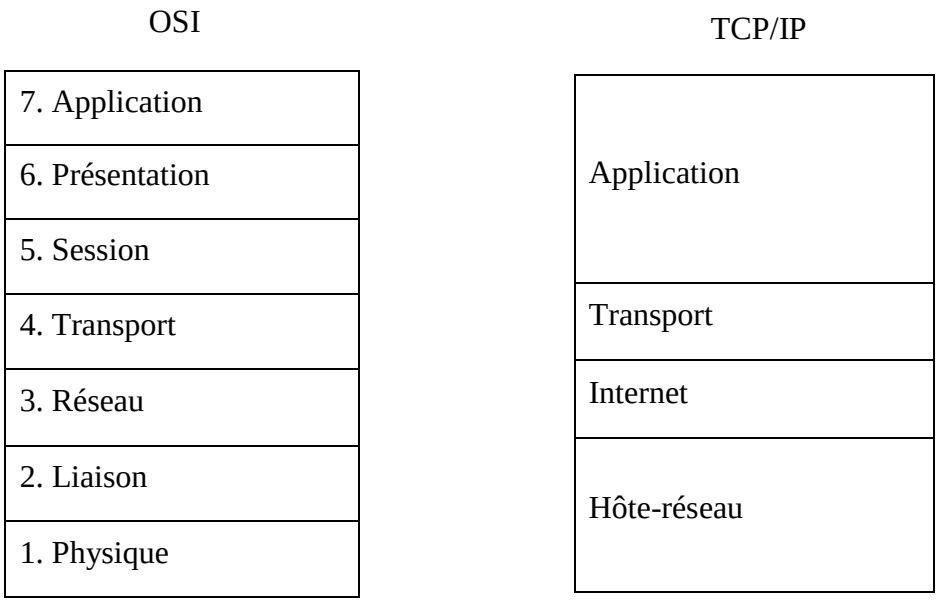



**Figure 8:** Le modèle OSI et le modèle TCP/IP

Les services de la couche 1 et 2 (Physique et Liaison) du modèle OSI sont intégrés dans une seule couche (hôte-réseau) ; les coucheS 5 et 6 (Session et Présentation) n'existent pas dans le modèle TCP/IP et leurs services sont réalisés par la couche Application. [6]

**a) La couche Hôte-réseau**

Elle regroupe tous les éléments nécessaires pour accéder à un réseau physique, quel qu'il soit. Elle contient en particulier les spécifications concernant la transmission de données sur le réseau physique, tout comme la première couche du modèle OSI.

**b) La couche Internet**

Elle permet aux hôtes d'envoyer des paquets élémentaires indépendants les uns des autres, sans se préoccuper des détails concernant leur acheminement vers l'hôte destination.

**c) La couche Transport**

Permet aux applications d'échanger des données indépendamment du réseau utilisé, grâce aux protocoles TCP et UDP.

**d) La couche Application**

Contient entre autres tous les protocoles de haut niveau, comme Telnet ou FTP. [1]

## Conclusion

Au cours de ce chapitre, nous avons pris connaissance des différents aspects liés à la sécurité des réseaux informatiques, les attaques qui menacent cette dernière et comment se protéger. La sécurité des systèmes informatiques est vitale à son bon fonctionnement. Il est donc nécessaire d'assurer sa protection, nous allons décrire dans le chapitre suivant les systèmes de détection d'intrusion.

## Introduction

De nos jours, les systèmes d'informations des entreprises subissent des différents attaques qui peuvent entrainer des pertes conséquentes, vue leurs évolution sur les plans d'échange d'informations d'une part et l'ouverture sur le monde extérieur d'autre part. Alors les systèmes de détection et prévention d'intrusion sont largement répandus pour la sécurité de ces systèmes informatiques puisqu' ils permettent à la fois de détecter et de répondre à une attaque en temps réel ou en hors-ligne.

En effet, dans ce chapitre nous allons d'abord présenter la notion de système de détection d'intrusion, ensuite nous présentons les systèmes de préventions d'intrusions et enfin une comparaison entre les deux.

## III.1. Systèmes de détection d'intrusion (IDS)

En 1980, James Anderson introduit un nouveau concept de système de détection d'intrusion qui ne connaîtrait un réel départ du domaine qu'en 1987 avec la publication d'un modèle de détection d'intrusion.

Par la suite, la recherche s'est développée et la détection d'intrusion est devenue une industrie mature et une technologie éprouvée, pratiquement tous les problèmes simples ont été résolus et aucune grande avancée n'a été effectuée dans ce domaine ces dernières années et les éditeurs de logiciels se concentrent plus sur la perfection des techniques de détection existantes. [5]

### III.1.1. Principes de fonctionnement des IDS

Un IDS est un équipement permettant de surveiller l'activité d'un réseau ou d'un hôte donné. Composé généralement de logiciel et éventuellement de matériel, ce système informatique a le rôle de détecter toute tentative d'intrusion. Par définition, un IDS n'a pas de signal antiseptique ou réactive dans la mesure où il n'empêche pas une intrusion de ce produire. Il se contente juste de faire une analyse de certaines informations en vue de détecter des activités malveillantes qui seront notifiées aux responsables de la sécurité du système dans le plus bref délai possible. Cette raison fait de la majorité des IDS des systémes qui opèrent en temps réel pourtant, il y'a des IDS qui réagissent en mettant fin a une connexion suspecte par exemple suite a la détection d'une intrusion. [5]

#### III.1.1.1. Principes de détection d'intrusion

Généralement, dans les systèmes informatiques, il existe deux types d'approches pour la détection d'intrusion : l'approche par scénario ou bien dite par signature basé sur un modèle constitué des actions interdites contrairement a l'approche comportementale qui est basé sur un modèle constitué des actions autorisées.

**A. Approche par scénario ou signature**

Dans cette approche, les détecteurs d'intrusion reposent sur la création d'une base de motifs qui représente des scénarios d'attaque connus au préalable et qui sera utilisé par la suite, le plus souvent en temps réel, sur les informations fournies par les sondes de détection. C'est donc un système de reconnaissance de motifs qui permet de mettre en évidence dans ces informations la présence d'une intrusion connue par la base de signature.

Les IDS à signature utilisent trois famille de méthodes qui se basent toutes sur la recherche d'un profil connu d'attaque. [9]

**a) Reconnaissance de forme (pattern matching)**

Cette méthode consiste à identifier une suite d'événements ou des marques d'une attaque connue dans les paquets analysés. En fait, le trafic réseau peut être vu comme une chaîne de caractères principale et les scénarios d'attaque comme des sous-suites qu'on veut identifier. [7]

**b) Système expert**

L'idée consiste à coder les attaques sous forme de règles condition-action où les conditions portent sur l'état du système surveillé ainsi que la nature des événements analysés, tandis que les actions permettent, soit de mémoriser le nouvel état du système, soit de conclure à la présence d'une attaque. [7]

**c) Algorithme génétique**

Les algorithmes génétiques utilisent la notion de sélection naturelle et l'appliquent a une population de solutions potentielles à un problème difficile, pour trouver une solution approchée dans un temps raisonnable. [7]

**B. Approche comportementale (Anomaly detection)**

Les détecteurs d'intrusion comportementaux reposent sur la création d'un modèle de référence qui représente le comportement de l'entité surveillé en situation de fonctionnement normale. Ce modèle est ensuite utilisé durant la phase de détection afin de pouvoir mettre en évidence d'éventuelles déviations comportementales. Pour cela, le comportement de l'entité surveillée est comparé à son modèle de référence. Le principe de cette approche est de considérer tout comportement n'appartenant pas au modèle de comportement normale comme une anomalie symptomatique d'une intrusion ou d'une tentative d'intrusion. [9]

**a) Approche probabiliste**

Des probabilités sont établies permettant de représenter une utilisation courante d'une application ou d'un protocole. Toute activité qui ne respecte pas le modèle probabiliste provoquera la génération d'une alerte. [15]

**b) Approche statistique**

Cette méthode consiste en un calcul de la variation (en fonction du temps) de l'utilisation d'un certain nombre de ressources. La constatation d'un changement dans la variation conduit au déclenchement d'une alerte. [14]

**c) Immunologie**

L'objet de cette méthode consiste à observer les services pendant un temps suffisamment représentatif de manière à établir une base des séquences d'appel normales. [15]

## III.2. Différents types d'IDS

Les attaques utilisées par les pirates sont très variées, puisque certaines utilisent des failles réseaux et d'autres des failles de programmation. C'est la raison pour laquelle la détection d'intrusion doit se faire à plusieurs niveaux.

Donc, on distingue deux d'IDS que nous détaillerons ci-dessous les caractéristiques principales.

### III.2.1. Systèmes de détection d'intrusion réseau (NIDS)

Les IDS réseaux analysent en temps réel le trafic qu'ils aspirent à l'aide d'une sonde (carte réseau en mode promiscuous[2]. Ensuite, les paquets sont décortiqués puis analysés.

Il est fréquent de trouver plusieurs IDS sur les différentes parties du réseau. On trouve souvent une architecture composée d'une sonde placée à l'extérieure du réseau afin d'étudier les tentatives d'attaques et d'une sonde en interne pour analyser les requêtes ayant traversé le pare-feu. [10]

### III.2.2. Systèmes de détection d'intrusion de type hôte (HIDS)

Les IDS systèmes analysent le fonctionnement de l'état des machines sur les quelles ils sont installés afin de détecter les attaques en se basant sur des démons (tels que syslogd [2] par exemple). L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levées. [10]

### III. 2.3. Systèmes de détection d'intrusion Hybrides

Les IDS hybrides rassemblent les caractéristiques des NIDS et HIDS. Ils permettent, en un seul outil, de surveiller le réseau et les terminaux. Les sondes sont placées en des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout, et lier les informations d'origine multiple. Ainsi, on comprend que les IDS hybrides sont basés sur une architecture distribuée, ou chaque composant unifie son format d'envoi. Cela permet de communiquer et d'extraire des alertes plus pertinentes. [10]

### III.2.4. Comparaison entre NIDS, HIDS et Systèmes hybrides

| | NIDS | HIDS | Systèmes hybrides |
|---|---|---|---|
| avantages | -contrôler un grand nombre d'hôtes avec un petit coût de déploiement.<br>-identifier les attaques de/à multiples hôtes.<br>-assurer la sécurité contre les attaques puisqu'il est invisible | -contrôler les activités locales des utilisateurs avec précision.<br>-capable de déterminer si une tentative d'attaque est couronnée de succès.<br>- capable de fonctionner dans des environnements cryptés | -moins de faux positifs.<br>-meilleure corrélation (la corrélation permet de générer de nouvelles alertes à partir de celles existantes).<br>-possibilité de réaction sur les analyseurs. |
| inconvénients | -incapable de fonctionner dans des environnements cryptés<br>-ne permet pas d'assurer si une tentative d'attaque est couronnée de succès. | -la difficulté de déploiement et de gestion, surtout lorsque le nombre d'hôte qui ont besoin de protection est large.<br>-incapable de détecter des attaques contre de multiples cibles dans le réseau. | |

**Tableau 1 :** la comparaison entre NIDS, HIDS et hybrides

## III.2. Systèmes 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 peut certes détecter des attaques contre un réseau mais ne peut empêcher l'intrusion. Cela a mené certaines entreprises utilisatrices à se poser la question : pourquoi s'investir dans une solution de détection des 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. [15]

### III.2.1. Systèmes de prévention d'intrusion réseau (NIPS)

#### III.2.1.1. Définition

Un NIPS est un logiciel ou matériel connecté directement à un segment du réseau.il a comme rôle d'analyser les tous les paquets circulant dans ce réseau. La principale différence entre un NIDS et un NIPS tient principalement en deux caractéristiques: le positionnement en coupure sur le réseau du NIPS et non plus seulement en écoute comme pour le NIDS et la

possibilité de bloquer immédiatement les intrusions et ce quelque soit le type de protocole de transport utilisé et sans reconfiguration d'un équipement tierce. Ce qui induit que le NIPS est constitué d'une technique de filtrage de paquets et de moyens de blocage.

### III.2.1.2. 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 pour le moment comme un IDS en déterminant si oui ou non le paquet est malveillant. Cependant, en plus de déclencher une alerte dans le cas ou il détecte un paquet suspect, il rejettera le paquet et marquera cette session suspecte. Quand les paquets suivants de cette session arriveront 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'. [2]

## III.3. Systèmes de prévention d'intrusion Kernel (KIPS)

Grâce à un KIPS, tout accès suspect peut être bloqué directement par le noyau, empêchant ainsi toute modification dangereuse pour le système. Le KIPS peut reconnaître des motifs caractéristiques du débordement de mémoire, et peut ainsi interdire l'exécution du code. Le KIPS peut également interdire l'OS d'exécuter un appel système qui ouvrirait un shell de commandes .Puisqu'un KIPS analyse les appels systèmes, il ralentit l'exécution. C'est pourquoi ce sont des solutions rarement utilisées sur des serveurs souvent sollicités. [3]

## III.4. Comparaison entre les IDS et les IPS

| Détection d'intrusion | Prévention d'intrusion |
|---|---|
| HIDS | HIPS |
| Le pour :<br>-alerte sur des changements au niveau du système.<br>-détection d'utilisation d'un système violant la politique de sécurité de l'entreprise.<br>Le contre :<br>-un cout élevé de déploiement et de la gestion<br>- une détection réussie vient d'une tentative réussie d'attaques. | Le pour :<br>-assurer la protection contre les attaques inconnues.<br>-exige peut ou aucune mise à jour dans une période annuelle.<br>-empêcher les attaques de s'exécuter sur une machine au niveau noyau que de détecter les résultats d'une attaque réussi<br>Le contre :<br>-le temps de déploiement peut être long afin d'équiper chaque serveur et/ou post de travail.<br>-le produit nécessite un ajustement après l'installation initial pour être outil de sécurité fonctionnel. |
| NIDS | NIPS |
| Le pour :<br>-capable de détecter des anomalies même sur les systèmes qui emploient le cryptage.<br>-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.<br>Le contre :<br>-à moins qu'un plan de réponse ne soit conçu et mis en place, l'IDS fournit peu ou aucune sécurité.<br>-un déploiement réussi demande un important ajustement de l'IDS pour réduire au minimum les faux positifs. | Le pour :<br>-peut arrêter la propagation des vers si déployé correctement sans arrêter le trafic.<br>-protège contre les nouvelles attaques avant que le code d'exploit soit sorti.<br>-réduire le cout de la réponse aux incidents.<br>Le contre :<br>-le cout du déploiement NIPS au sein d'un réseau peut être important.<br>-un NIPS nécessite toujours des mises à jour de sécurité pour être vraiment efficace. |

**Tableau 2 :** la comparaison entre IDS et IPS

## III.5. Inconvénients des IDS/IPS

Les systèmes de détection et de prévention d'intrusions représentent des moyens nécessaires de la sécurité des systèmes informatique. Mais ils restent insuffisants tant qu'il représente des inconvénients. Nous allons détailler quelques uns en ce qui suit :

### III.5.1. Besoin de connaissances en sécurité

La mise en place de sonde sécurité fait appel à de bonnes connaissances en sécurité. L'installation en elle-même des logiciels est à la portée de n'importe quel informaticien. En revanche l'exploitation des remontées d'alertes nécessite des connaissances plus pointues. Les interfaces fournissent beaucoup d'informations, et permettent des tris facilitant beaucoup le travail, mais l'intervention humaine est toujours indispensable. A partir des remontées d'alertes, quelle mesure prendre ? Est-il utile de relever des alertes dont toutes les machines sont protégées? Et Comment distinguer un faux-positif d'un véritable incident de sécurité ? Toutes ces questions et bien d'autres doivent se poser au responsable de sécurité en charge d'un IDS. La configuration, et l'administration des IDS nécessitent beaucoup de temps, et de connaissances. C'est un outil d'aide, qui n'est en aucun cas complètement automatisé. [2]

### III.5.2. Faux positifs et faux négatifs

La détection d'anomalie est capable de détecter les attaques inconnues ; toutefois, elle n'est pas aussi efficace que la détection d'abus pour les attaques connues. Notamment, un fort taux de faux positifs peut être rencontré si le paramétrage de l'IDS n'a pas été réalisé avec soin. [2]

### III.5.3. Pollution / Surcharge

Les IDS peuvent être pollués ou surchargés, par exemple par la génération d'un trafic important (le plus difficile et lourd possible à analyser). Une quantité importante d'attaques peut également être envoyée afin de surcharger les alertes de l'IDS. Des conséquences possibles de cette surcharge peuvent être la saturation de ressources (disque, CPU, mémoire), la perte de paquets, le déni de service … [12]

### III.5.4. Consommation de ressources

La détection d'intrusion est excessivement gourmande en ressources. En effet un système NIDS doit générer des journaux de comptes-rendus d'activité anormale ou douteuse sur le réseau. [2]

### III.5.5. Perte de paquets (limitation des performances)

Les vitesses de transmission sont parfois telles qu'elles dépassent largement la vitesse d'écriture des disques durs, ou même la vitesse de traitement des processeurs. Il n'est donc pas rare que des paquets ne soient pas traités par l'IDS, et que certains d'entre eux soient néanmoins reçus par la machine destinataire. [12]

## Conclusion

Ce chapitre nous a permis de découvrir les systèmes de détection d'intrusion leurs fonctionnement et leurs capacités et il nous est paru évident que ces systèmes sont à présent indispensables aux entreprises afin d'assurer leur sécurité informatique, compléter et aide les tâches des autres équipements de sécurité. Nous allons voir dans le chapitre suivant comment réussir une configuration de ces derniers afin de mieux sécurisé le réseau.

## Introduction

Comme nous avons vu dans le chapitre qui précède, des IDS et IPS ont été proposés pour la sécurisation des réseaux. L'objectif principal de ce chapitre est de présenter notre contribution dans le cadre de ce mémoire.

Dans ce qui suit, nous commencerons d'abord par une description générale de l'outil SNORT. Ensuite nous discuterons de l'environnement utilisé pour l'implémentation de la solution, ainsi que son emplacement dans le réseau. Enfin nous terminerons avec la présentation des différentes manipulations (installations, configurations et fonctionnalités).

## IV.1. Présentation générale de Snort

En anglais, Snort signifie «renifler ». Snort est un système de détection d'intrusion libre (ou NIDS) publié sous licence GNU GPL. À l'origine écrit par Martin Roesch, il appartient actuellement à Sourcefire. Des versions commerciales intégrant du matériel et des services de supports sont vendues par Sourcefire. Snort est un des NIDS les plus performants. Il est soutenu par une importante communauté qui contribue à son succès.

### IV.1.1. Fonctionnement

Snort capture des paquets sur un point d'un réseau IP, analyse le flux obtenu en temps réel, et compare le trafic réseau à une base de données d'attaques connues. Les attaques connues sont répertoriées dans des librairies de règles mises à jour par plusieurs communautés très actives

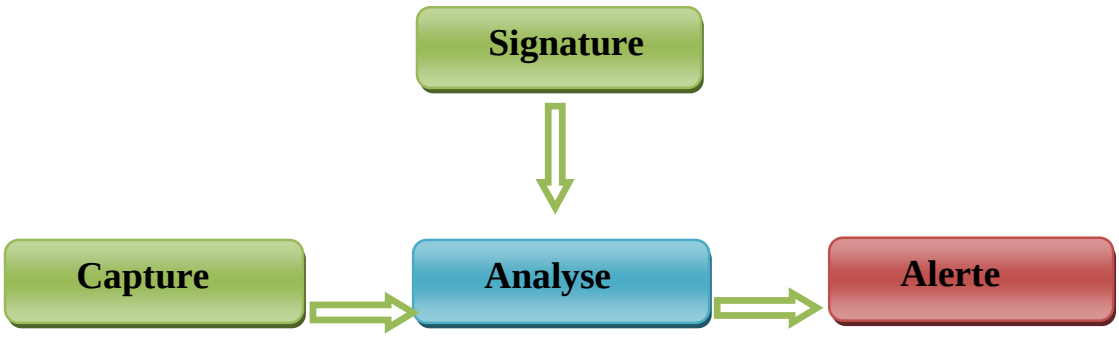



**Figure 9:** Le fonctionnement de snort

Snort peut également être utilisé avec d'autres modules compatibles (tels que des interfaces graphiques, des actualisateurs de librairies d'attaques indépendants, etc.)
Snort est compatible avec la plus part des OS. Windows, Mac, Linux Ubuntu, CentOS… [17]

### IV.1.2. Positionnement de Snort dans un réseau

L'emplacement physique de la sonde SNORT sur le réseau a un impact considérable sur son efficacité. Dans le cas d'une architecture classique, composée d'un Firewall et d'une DMZ, trois positions sont généralement envisageables :

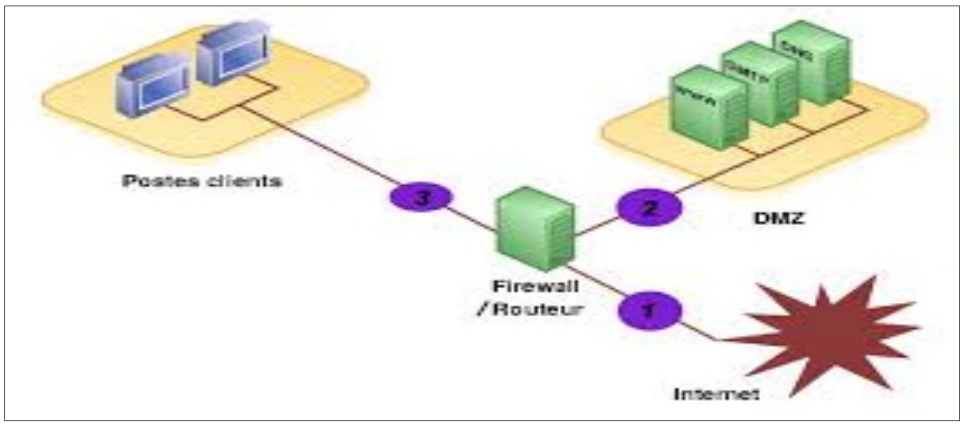



**Figure 10 :** les différentes positions de Snort dans un réseau

**a)** **Avant le Firewall ou le routeur filtrant :** dans cette position, la sonde occupe une place de premier choix dans la détection des attaques de sources extérieures visant l'entreprise. SNORT pourra alors analyser le trafic qui sera éventuellement bloqué par le Firewall. Les deux inconvénients de cette position du NIDS sont:

- Le risque engendré par un trafic très important qui pourrait entraîner une perte de fiabilité.
- Etant situé hors du domaine de protection du firewall, le NIDS est alors exposé à d'éventuelles attaques pouvant le rendre inefficace.

**b)** **Sur la DMZ :** dans cette position, la sonde peut détecter tout le trafic filtré par le Firewall et qui a atteint la zone DMZ. Cette position de la sonde permet de surveiller les attaques dirigées vers les différents serveurs de l'entreprise accessibles de l'extérieur.

**c)** **Sur le réseau interne :** le positionnement du NIDS à cet endroit nous permet d'observer les tentatives d'intrusion parvenues à l'intérieur du réseau d'entreprise ainsi que les tentatives d'attaques à partir de l'intérieur. Dans le cas d'entreprises utilisant largement l'outil informatique pour la gestion de leur activités ou de réseaux fournissant un accès à des personnes peu soucieuses de la sécurité (réseaux d'écoles et d'universités), cette position peut revêtir un intérêt primordial. [18]

### IV.1.2.1. La position de SNORT choisit

Vue que l'entreprise CEVITAL s'interesse aux menaces externes provenant du réseau internet, nous avons opté pour la première position (avant le firewall ou le routeur filtrant), ayant une caractéristique qui répond aux besoins de l'entreprise.

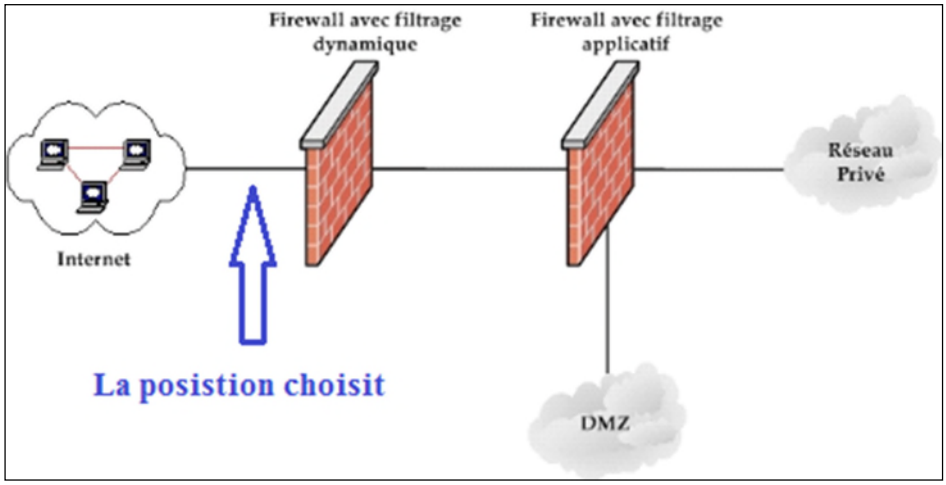



**Figure 11 :** La position de SNORT choisie

### IV.1.3. Architecture de SNORT

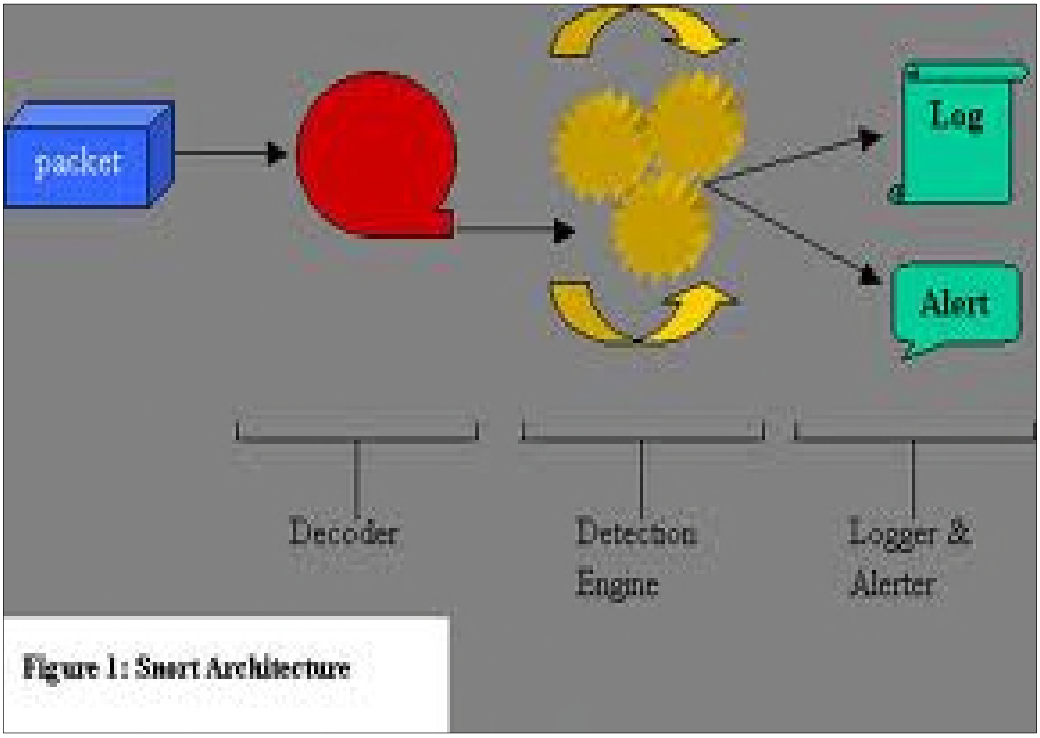



**Figure 12 :** Architecture de Snort

L'architecture de SNORT est modulaire (Figure 11), elle est composée de :

- **Un noyau de base :** (Packet Decoder) au démarrage, ce noyau charge un ensemble de règles, compile, optimise et classe celles-ci. Durant l'exécution, le rôle principal du noyau est la capture de paquets.
- **Une série de pré – processeurs :** ceux-ci améliorent les possibilités de SNORT en matière d'analyse et de recomposition du trafic capturé. Ils reçoivent les paquets directement capturés, éventuellement les retravaillent puis les fournissent au moteur de recherche de signatures.
- **Un ou plusieurs moteurs de détection (Detection Engine)** applique une série d'analyses aux paquets, ces analyses se composent principalement de comparaisons de différents champs des headers des protocoles (IP, ICMP, TCP et UDP) par rapport à des valeurs précises.

Après la détection d'intrusion, 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.

### IV.1.4. Environnement

L'entreprise nous a exigé de travailler dans un environnement Linux, plus précisément : CentOS-6.6, car il nous fournit un espace de travail unique et nous assure une fiabilité de résultats incompatible.

CentOS est une distribution GNU/Linux principalement destiné aux serveurs. Elle est l'une des distributions Linux les plus populaires pour les serveurs web. Depuis novembre 2013, elle est la troisième distribution la plus utilisée sur les serveurs web.


### IV.1.5. Installation de snort

Pour initialiser SNORT sous CentOS, nous devons d'abord installer les outils de compilations et les dépendances de Snort :

- Libpcap (Packet CAPture) : Librairie utilisée par Snort pour capturer les paquets.
- Libnet : c'est une bibliothèque logicielle open source. Elle permet de fabriquer et d'injecter facilement des paquets sur un réseau.
- GCC (GNU Compiler Collection) : un compilateur sous linux permettant de compiler du c, du c++, du java… indispensable pour compiler les sources de Snort.
- Libpcre : est une librairie de fonctions utilisant la même syntaxe et sémantique que Perle 5.

- Daq : permet d'acquérir des paquets sur le réseau. Indispensable pour les versions de snort après la 2.9.0
- Zlib : bibliothèque logicielle de compression de données.

### IV.1.6. Mode de fonctionnement

Il peut être configuré pour fonctionner en plusieurs modes :

- le mode sniffer : dans ce mode, SNORT lit les paquets circulant sur le réseau et les affiche d'une façon continue sur l'écran ;
- Le mode « packet logger » : dans ce mode SNORT journalise le trafic réseau dans des répertoires sur le disque ;
- le mode détecteur d'intrusion réseau (NIDS) : ce mode fait l'objet de notre stage. 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. [18]

### IV.1.7. Paramètre de snort

#### IV.1.7.1. 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>.

**Exemple :**

Preprocessor minfrag : 128.

#### IV.1.7.2. Les plugins de sortie

Les plugins de sortie ont été introduits dans la version 1.6. Ils permettent à Snort d'être plus souple dans la mise en forme et la présentation de la sortie aux utilisateurs. Les modules de sortie fonctionnent lorsque les sous-systèmes d'alerte ou de logging de Snort sont appelés, après les préprocesseurs et l'engin de détection. Comme les systèmes standards de logs et d'alertes, les modules de sortie envoient leurs données dans /var/log/snort par défaut ou un répertoire utilisateur spécifié avec l'option –l en ligne de commande.

Les modules de sortie sont chargés lors de l'exécution en spécifiant le mot clef de sortie : output <name> : <options>. [19]

#### IV.1.7.3. Les règles de snort

**a) 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. [19]

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

**Figure 13 :** les différents champs d'une règle SNORT.

➢ **Header**

- **Le champ « action » :** il peut prendre plusieurs valeurs selon l'action à mener par Snort en détectant des paquets réseaux répondant au critère définie 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.

➢ **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 numero 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 debut 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).

#### b) 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.2. Mise en place de 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 (Fast Unified Logging) et ses derniers seront exploités par Barnyard pour une inscription en base de données.

### IV.2.1. 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.3. 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 BASE (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, BASE 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.4. L’outil d’attaque

Pour tester la fiabilité de notre application, nous avons utilisé l’outil d’attaque LOIC, c’est une application de test de réseau, écrite en C# et développée par Praetox Technologies. Cette application tente d'attaquer par déni de service le site ciblé en inondant le serveur avec des paquets TCP, des paquets UDP, dont des requêtes HTTP avec l'intention de perturber le service d'un hôte particulier.

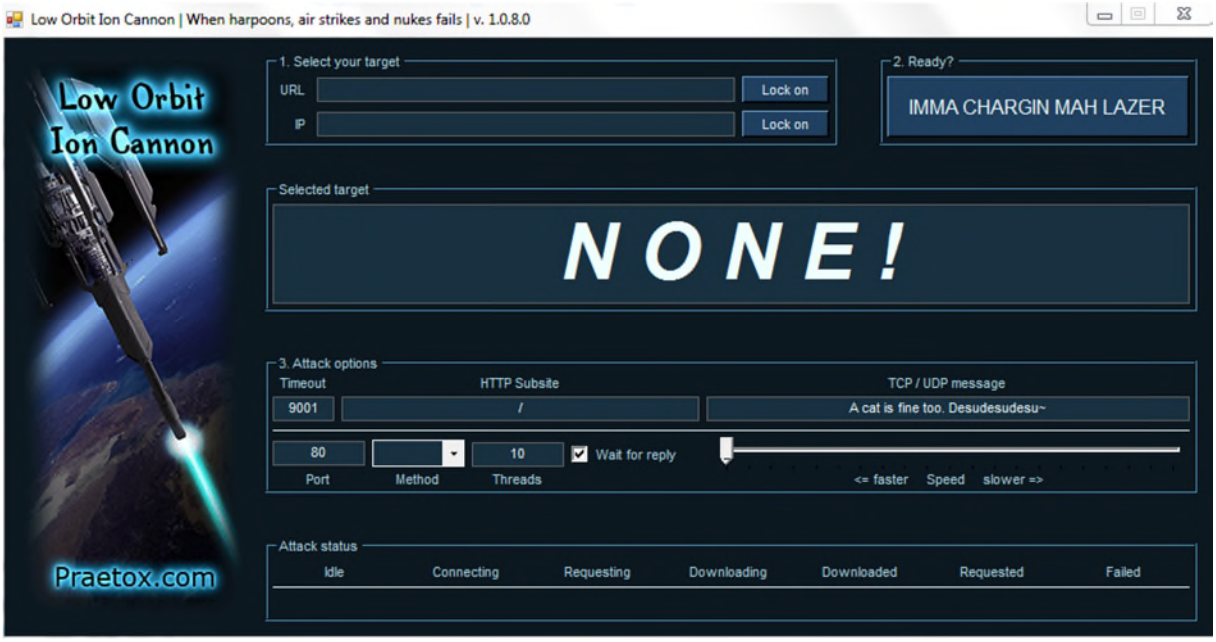



Figure 14 : l’interface de l’outil d’attaque LOIC

### IV.4.1.Principe de fonctionnement

Pour pouvoir attaquer divers sites, le fonctionnement de Loic est très simple : le logiciel déclare l'ordinateur comme robot au sein d'un réseau (botnet) piloté par un serveur maître. Ce système est classique, mais se fait en général à l'insu du propriétaire de la machine.

## Conclusion

Snort est un outil très intéressant dans la mise en place d’une sécurité réseau. Grace aux communautés très actives qui créent les librairies d’attaque, Snort permet de voir avec une bonne acuité de quoi il faut se protéger. Il est à souligner l’importance d’une bonne mise a jour de ces librairies. De plus Snort placé dans l’enceinte d’un réseau permet de détecter les failles les plus répandus qui proviennent généralement de l’exterieur.

## Introduction

Dans ce dernier chapitre, nous allons voir un cas pratique concernant snort, nous allons voir comment installer les différents composants du NIDS ainsi que toutes les configurations nécessaires.

Au final, nous allons tester notre configuration en lançant quelques attaques et essayer de les détecter.

## V.1. Mise en place de SNORT

### V.1.1. Installation de SNORT

On peut décomposer l'installation en deux parties :

- L'installation de l'outil SNORT :

| Commandes | Remarques |
|---|---|
| Cd/ usr/ local/ snort | Accéder au repértoire SNORT |
| Tar –xvzf snort-2.9.7.2.tar.gz | Décompresser l'application |
| ./configure | Configuration |
| Make | Compilation |
| Make install | Installation |

**Tableau 3 :** Les étapes de l'installation de SNORT

- L'installation des règles Snort

| **Commandes** | **Remarques** |
|---|---|
| Mkdir /etc/snort | Création du répertoire contenant la configuration Snort |
| Cp /usr/src/snort/snort.conf /etc/snort/ | Copie du fichier de configuration snort dans /etc/snort |
| Cp snortrules-snapshot-2972.tar.gz /etc/snort/ | Mise en place des règles dans le répertoire de configuration snort |
| Cd /etc/snort/ | On se place dans le répertoire de configuration Snort |
| Tar –xvzf snortrules-snapshot-2972.tar.gz | Décompactage des règles |

**Tableau 4 :** Les étapes de l'installation des règles SNORT

### V.1.2. Lancement de snort

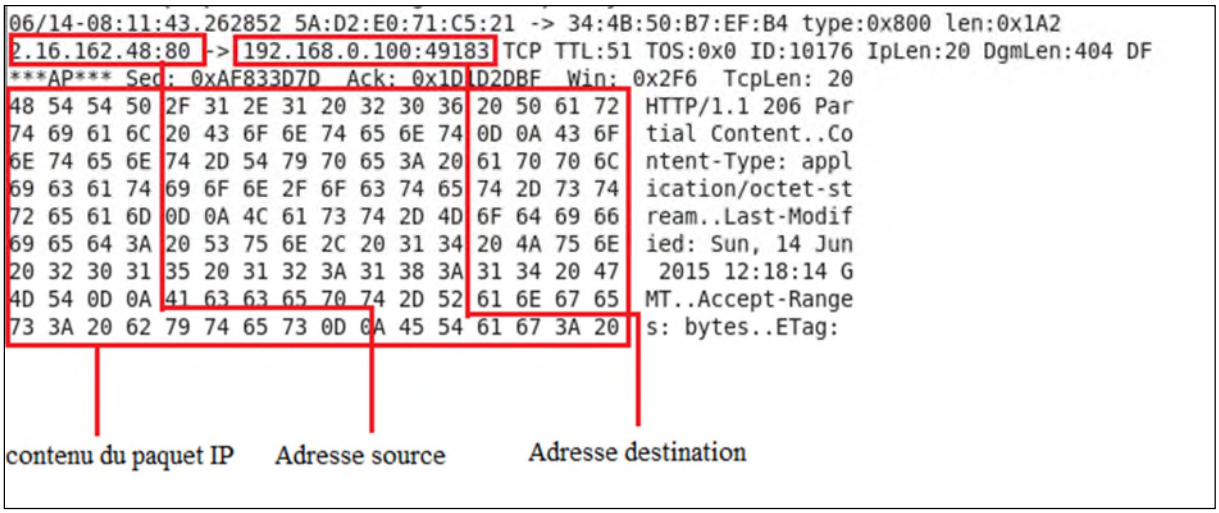
Vérification de l'installation de snort en tapant la commande suivante dans le terminal:
#snort –V


Si SNORT est bien installé, son lancemant avec la commande
#service snortd start nous donnera le resultat suivant


### V.1.3. Mode de fonctionnement

#### a) Le mode écoute (sniffer mode)

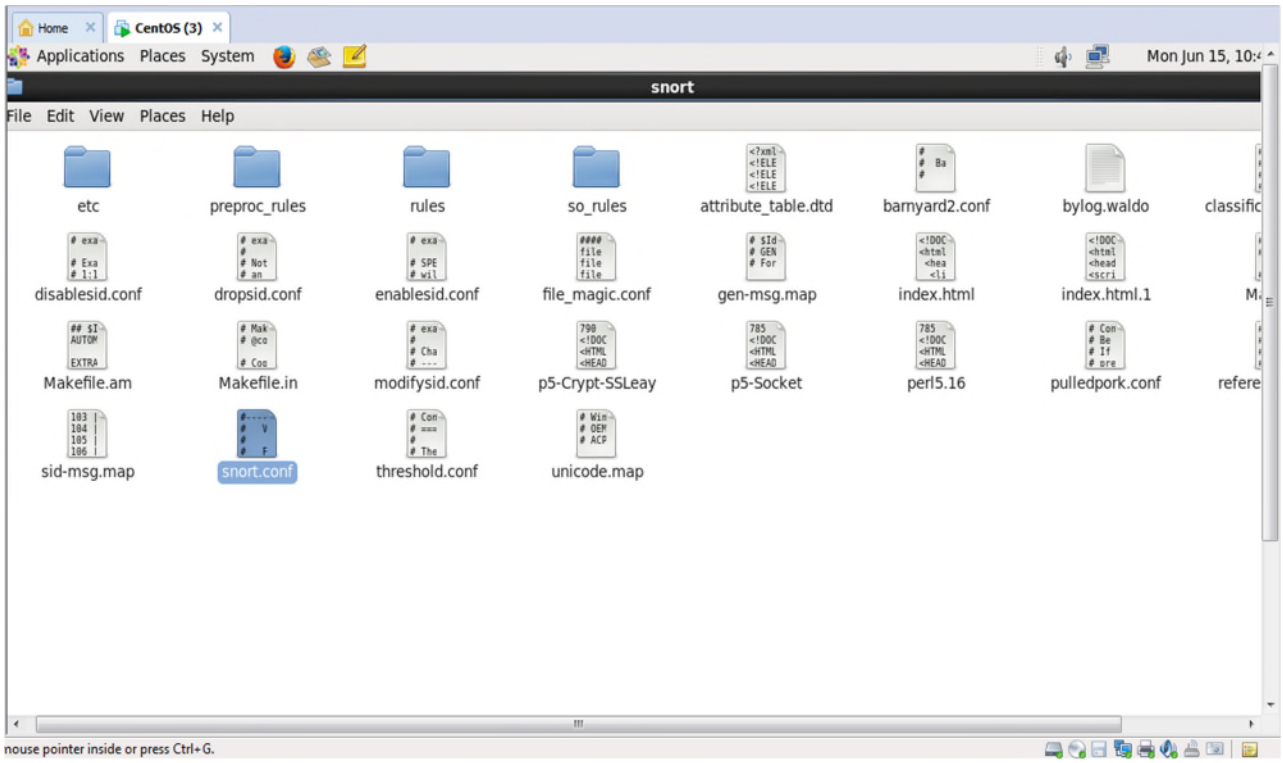
C'est le mode basic, il permet de lire et afficher les paquets TCP/IP circulant sur le réseau, d'une façon continue sur l'écran.


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

- **Configuration du fichier snort.conf**

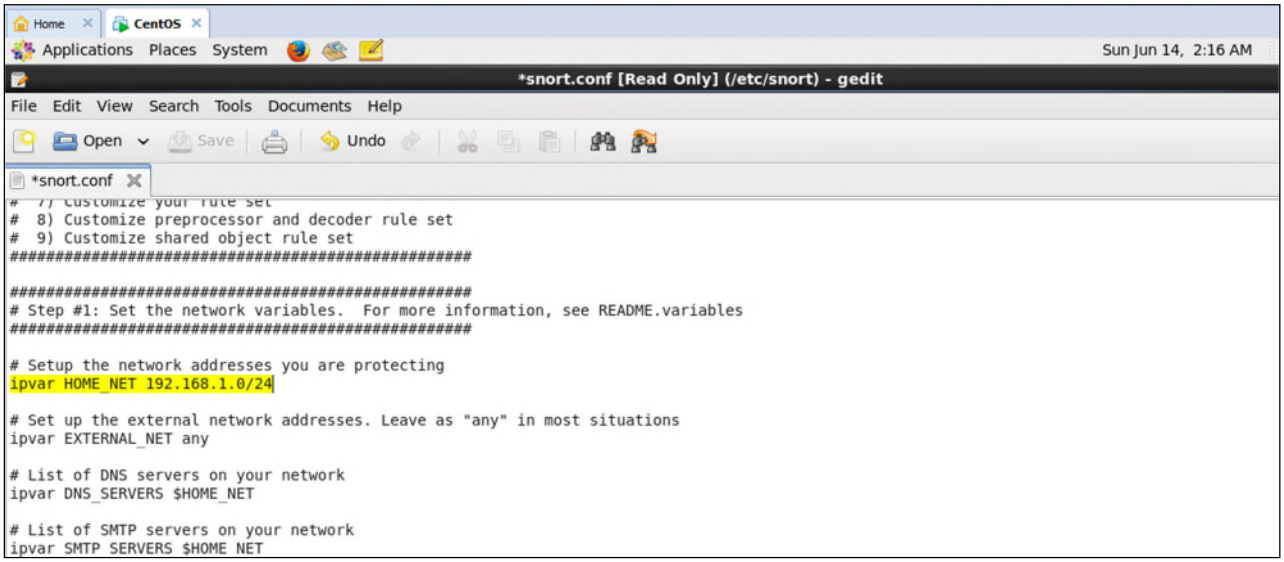



Nous pouvons modifier le fichier directement par le terminal en tapant la commande :

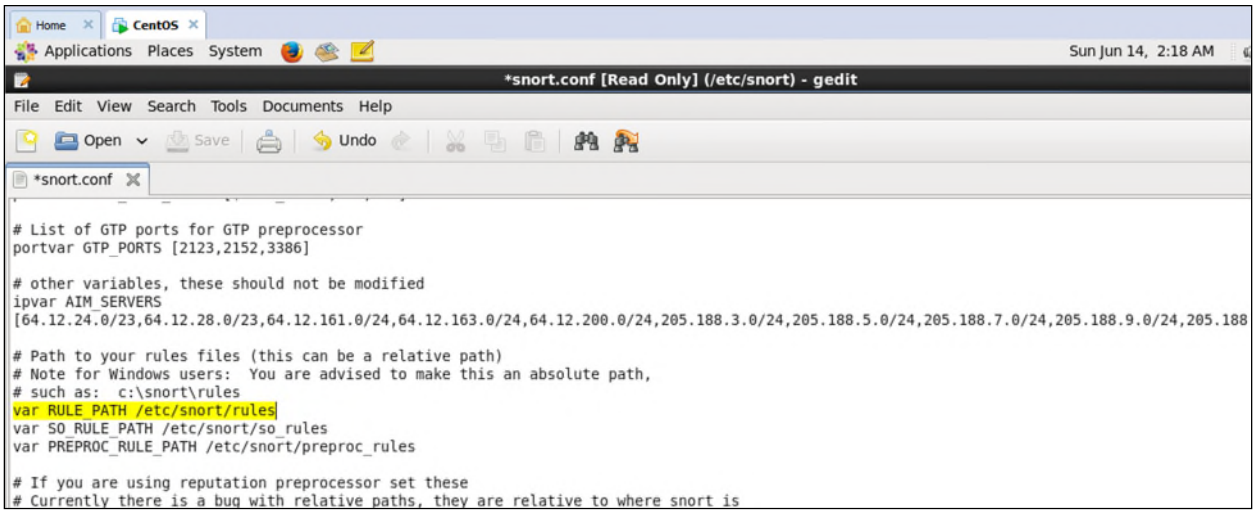
# vi /etc/snort/snort.conf

Ou accéder manuellement au fichier et le modifier.

Nous devons indiquer la classe d'adresse du réseau comme suite :

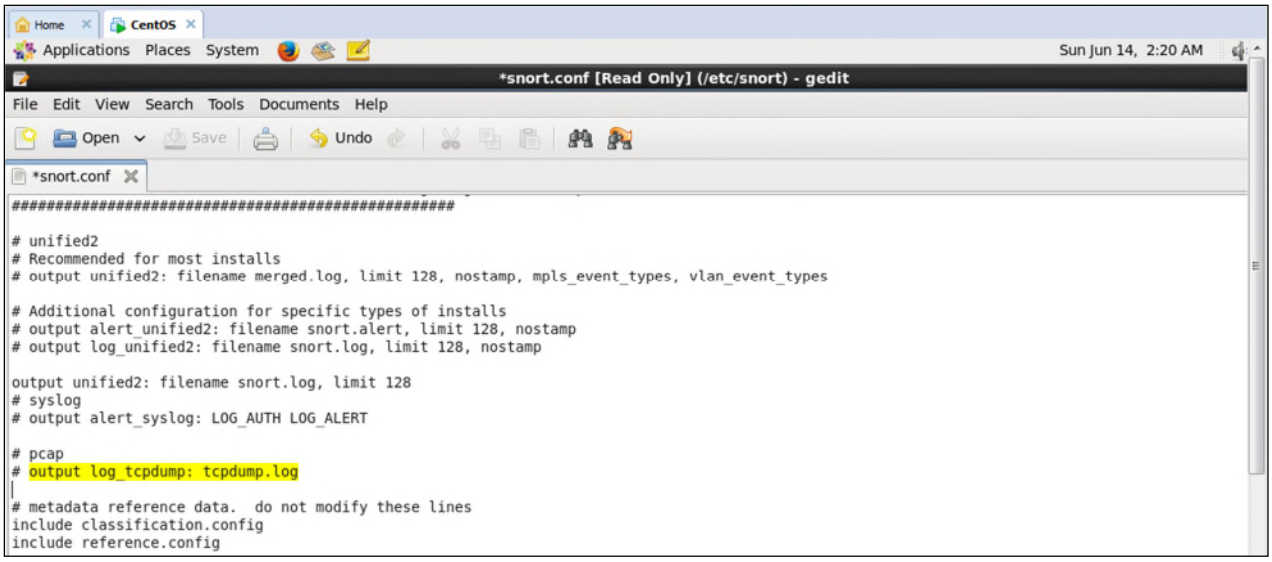


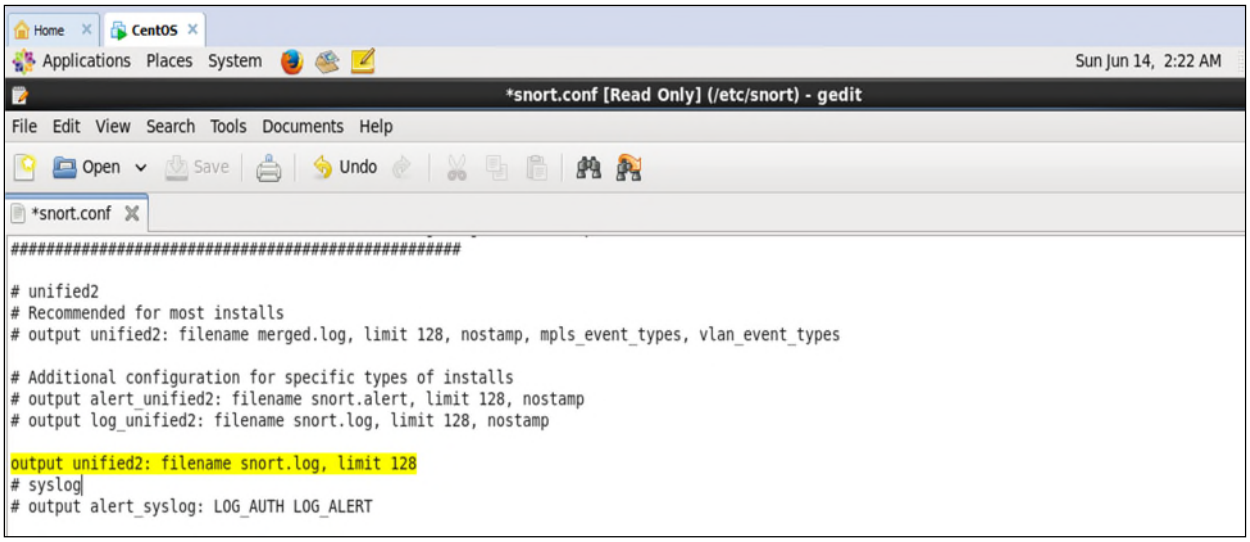
Puis il faudra spécifier le répertoire ou sont disposées nos règles (rules) :




On commente le fichier de sortie output log_tcpdump en ajoutant le caractère « # », car dans notre cas, on travail avec un module de sortie du format unified en ajoutant cette ligne :

Output unified2: filename snort.log, limit 128






## V.2. Mise en place de Barnyard2

### V.2.1. Installation de Barnyard2

A partir du terminal, on éxecute les commandes suivantes :

#wget https://www.github.com/firnsy/barnyard2/archive/v2-1.13.tar.gz
#tar zxvf v2-1.13.tar.gz
#cd barnyard2-2.1.13
#autoreconf –fvi –I ./m4
#./configure –with-mysql
#make
#make install

```
[root@localhost barnyard2-2-1.13]# make ; make install
make  all-recursive
make[1]: Entering directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13'
Making all in src
make[2]: Entering directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src'
Making all in sfutil
make[3]: Entering directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src/sfutil'
make[3]: Nothing to be done for `all'.
make[3]: Leaving directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src/sfutil'
Making all in output-plugins
make[3]: Entering directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src/output-plugins'
make[3]: Nothing to be done for `all'.
make[3]: Leaving directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src/output-plugins'
Making all in input-plugins
make[3]: Entering directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src/input-plugins'
make[3]: Nothing to be done for `all'.
make[3]: Leaving directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src/input-plugins'
make[3]: Entering directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src'
make[3]: Nothing to be done for `all-am'.
make[3]: Leaving directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src'
make[2]: Leaving directory `/usr/local/src/firnsy-barnyard2/barnyard2-2-1.13/src'
Making all in etc
```

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

# cp etc/barnyard2.conf /etc/snort

Ensuite on crée un dossier ou Barnyard2 stocke les logs :

# mkdir /var/log/barnyard2

### V.1.2. Lancement de Barnyard2

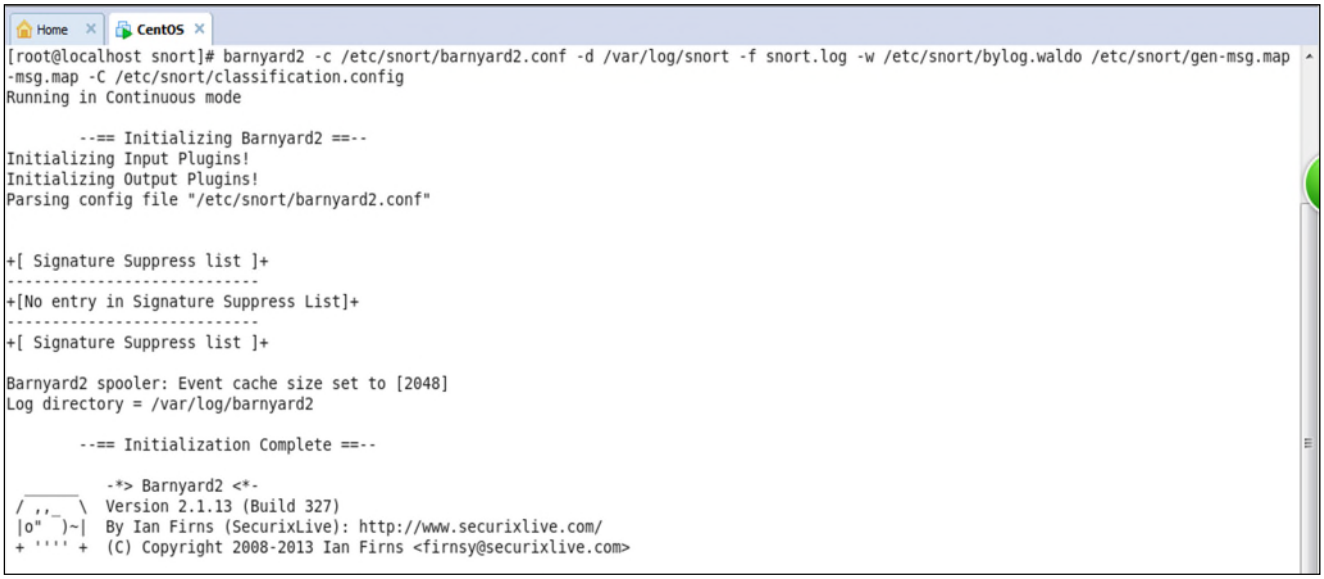
Apres l'installation, on vérifié si tout est bon en tapant la version du Barnyard :

#barnyard –V


Si Barnyard est bien installé, son lancemant avec la commande

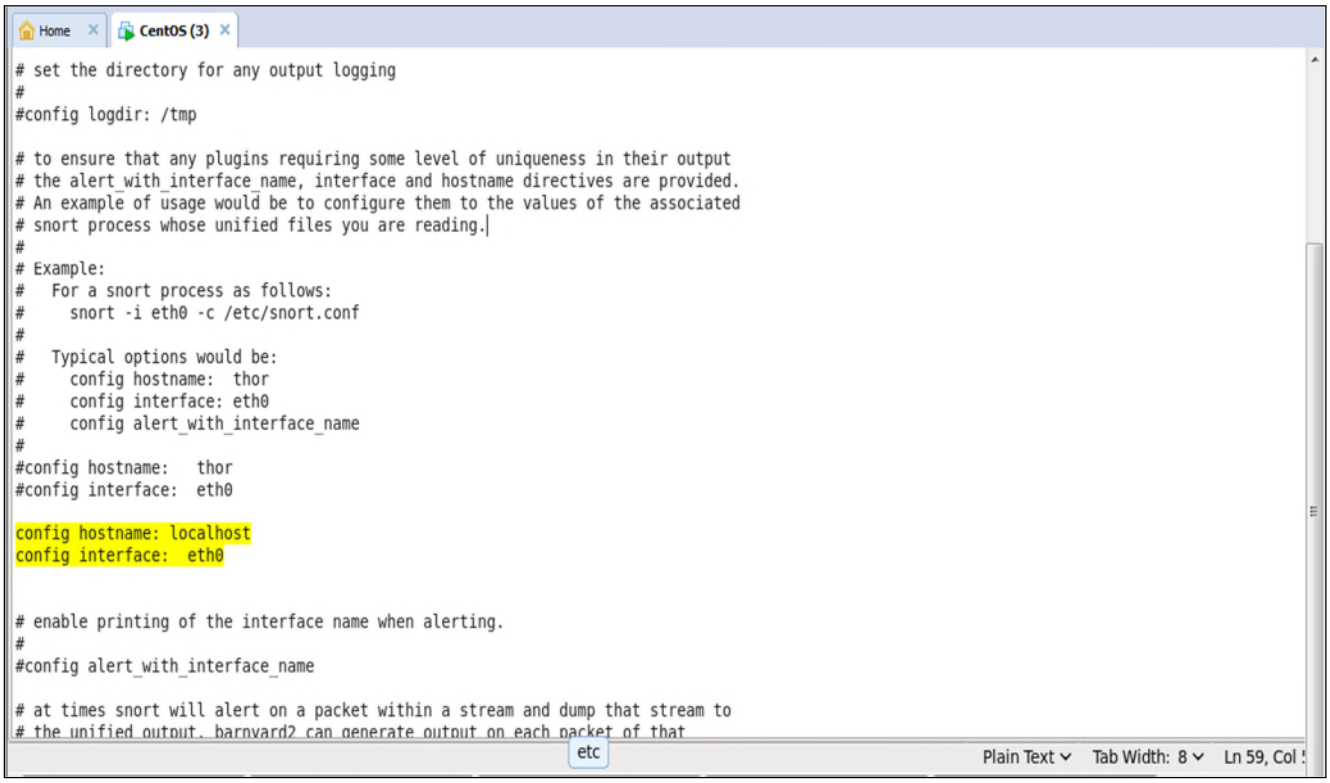
#barnyard2 –c /etc/snort/barnyard2.conf –d /var/log/snort –f snort.log –w /etc/snort/bylog.waldo /etc/snort/gen-msg.map /etc/snort/sid-msg.map –C /etc/snort/classification.config

Nous donnera le resultat suivant

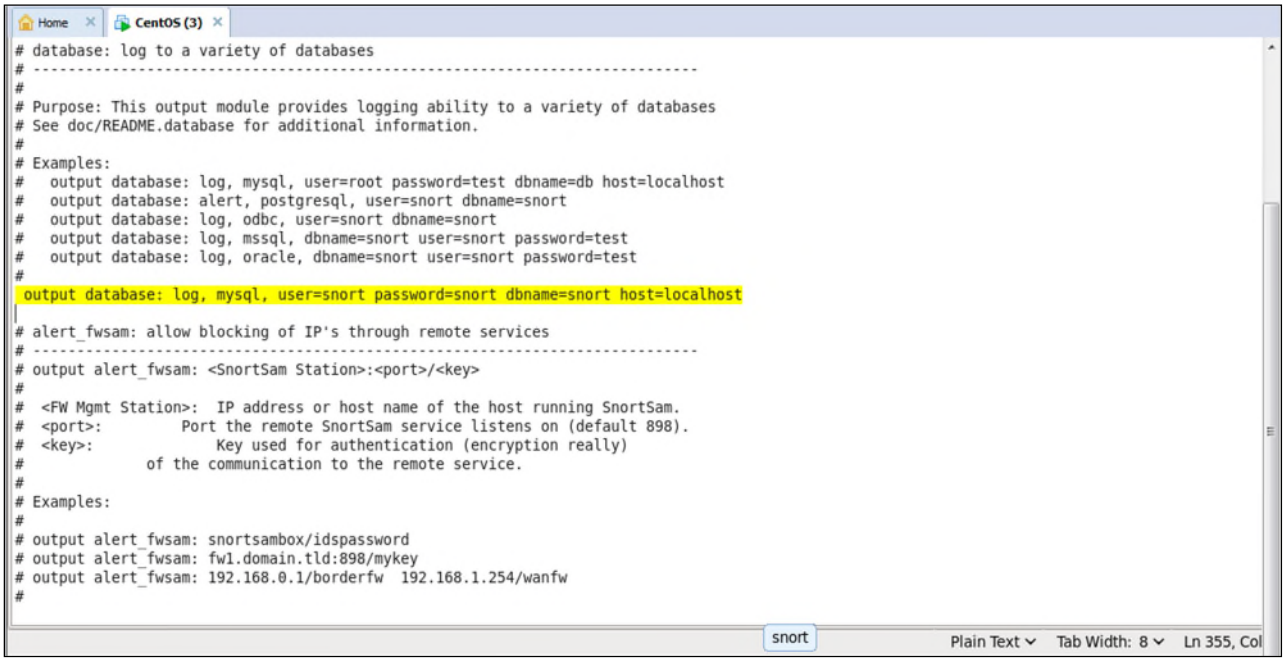


- **Configuration du fichier baryard2.conf**

Nous devons ajouter le nom du hôte ‘localhost’ est l’interface ‘eth0’ comme suite :

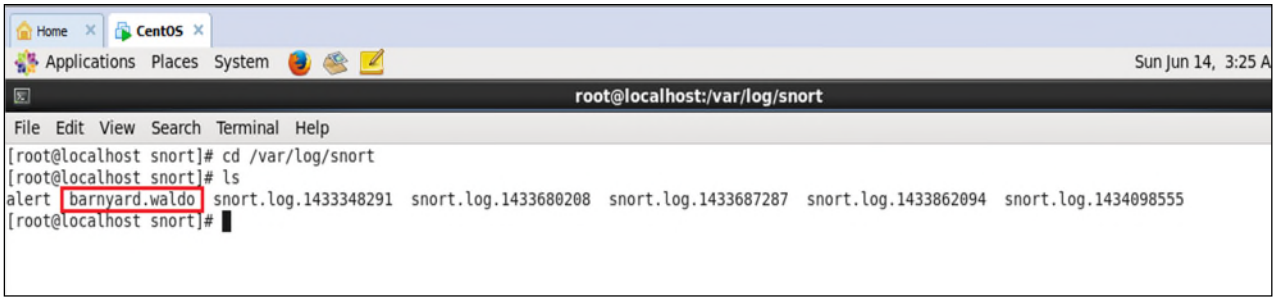



Ensuite nous devons configurer la sortie vers la base de données MySql

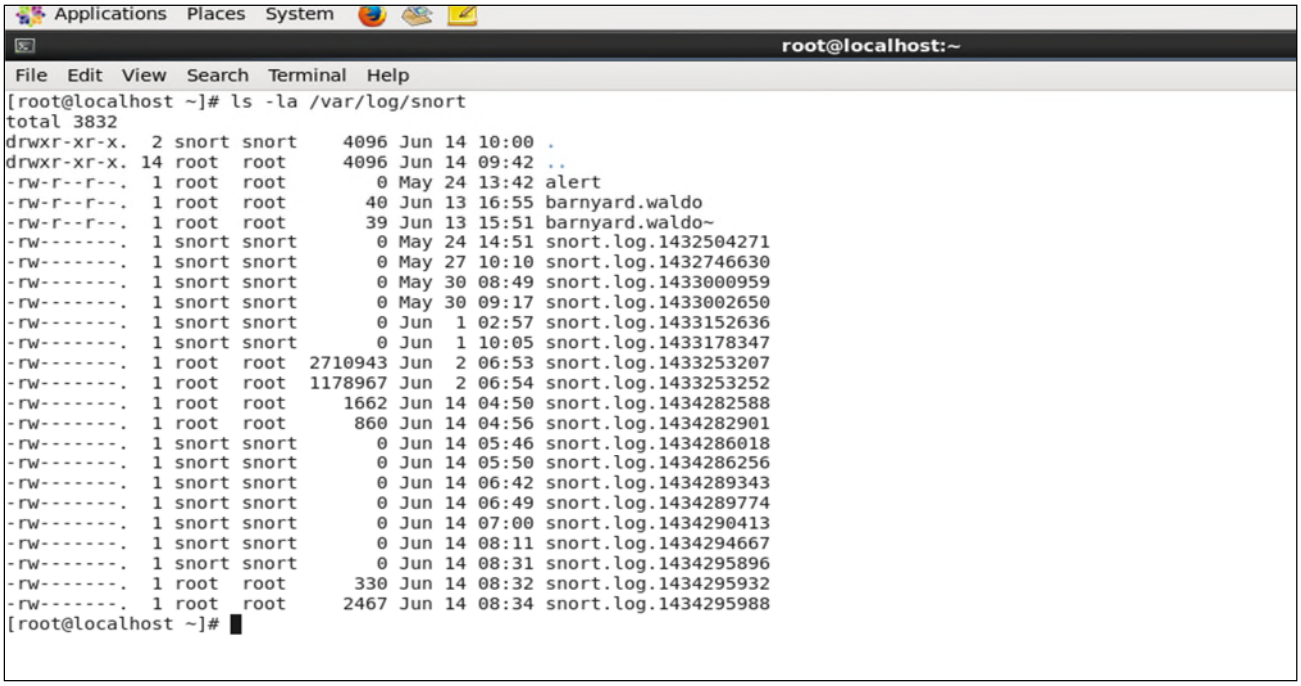


Cette étape a pour but de synchroniser Snort avec Barnyard : On crée un fichier portant le nom de barnyard.waldo dans le répertoire :

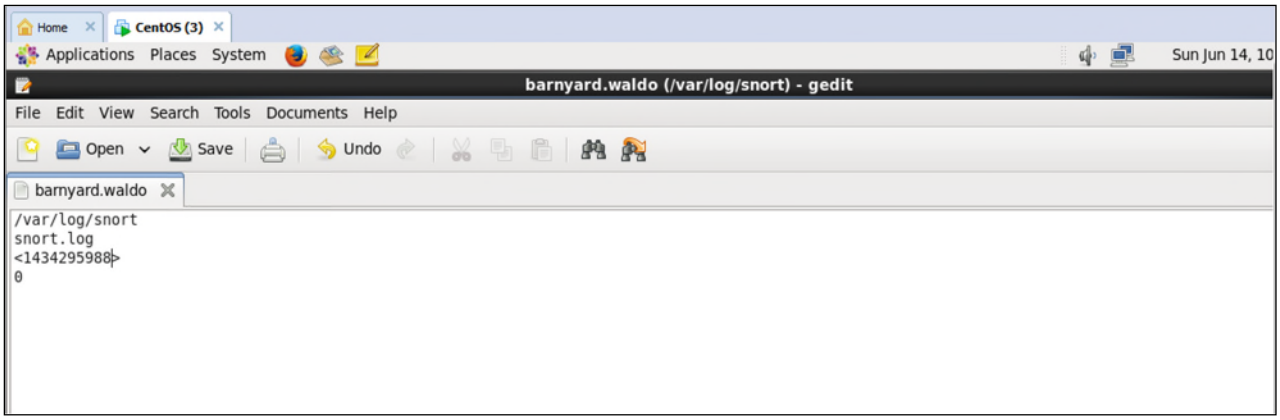
/var/log/snort




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



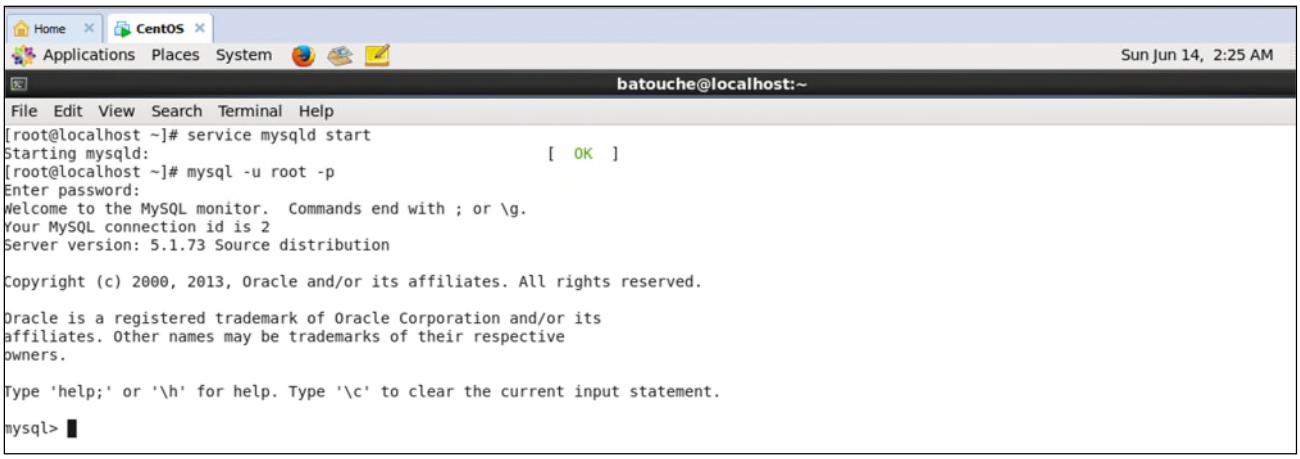

Puis on l’injecter dans le fichier barnyard.walo




## V.3. Mise en œuvre de la base de données MySQL

### V.3.1. Installation

La première étape consiste à installer MySQL-server et MySQL-devel :
#yum install mysql-server mysql-devel
On doit demarré 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 :




### V.3.2. Création de la base de données pour snort

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

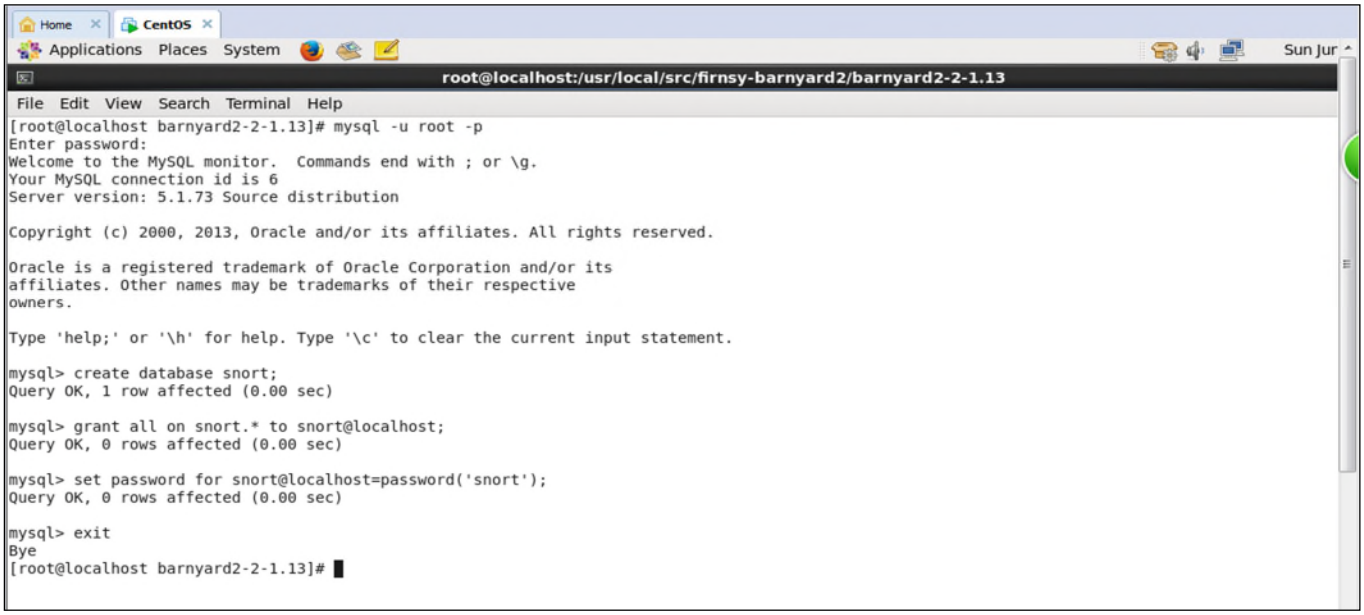
```
Mysql> create database snort;
```

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

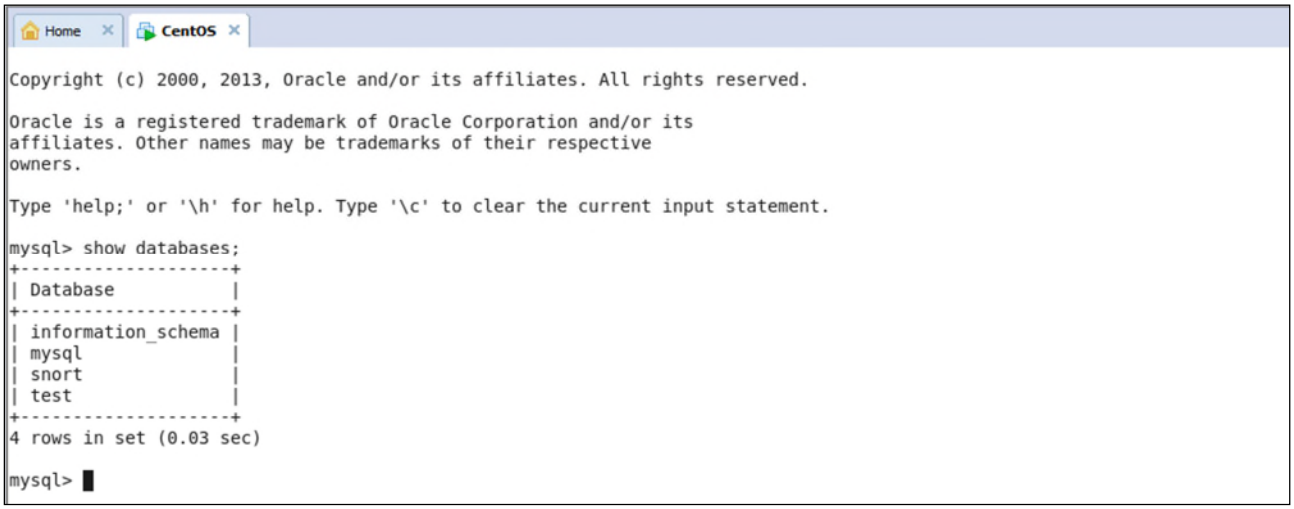
```
Mysql> grant all on snort.* to snort@localhost;
Mysql>set password for snort@localhost=password('snort');
```

Puis on recharge les privilèges MySQL:

```
Mysql> flush privileges ;
Mysql> exit
```

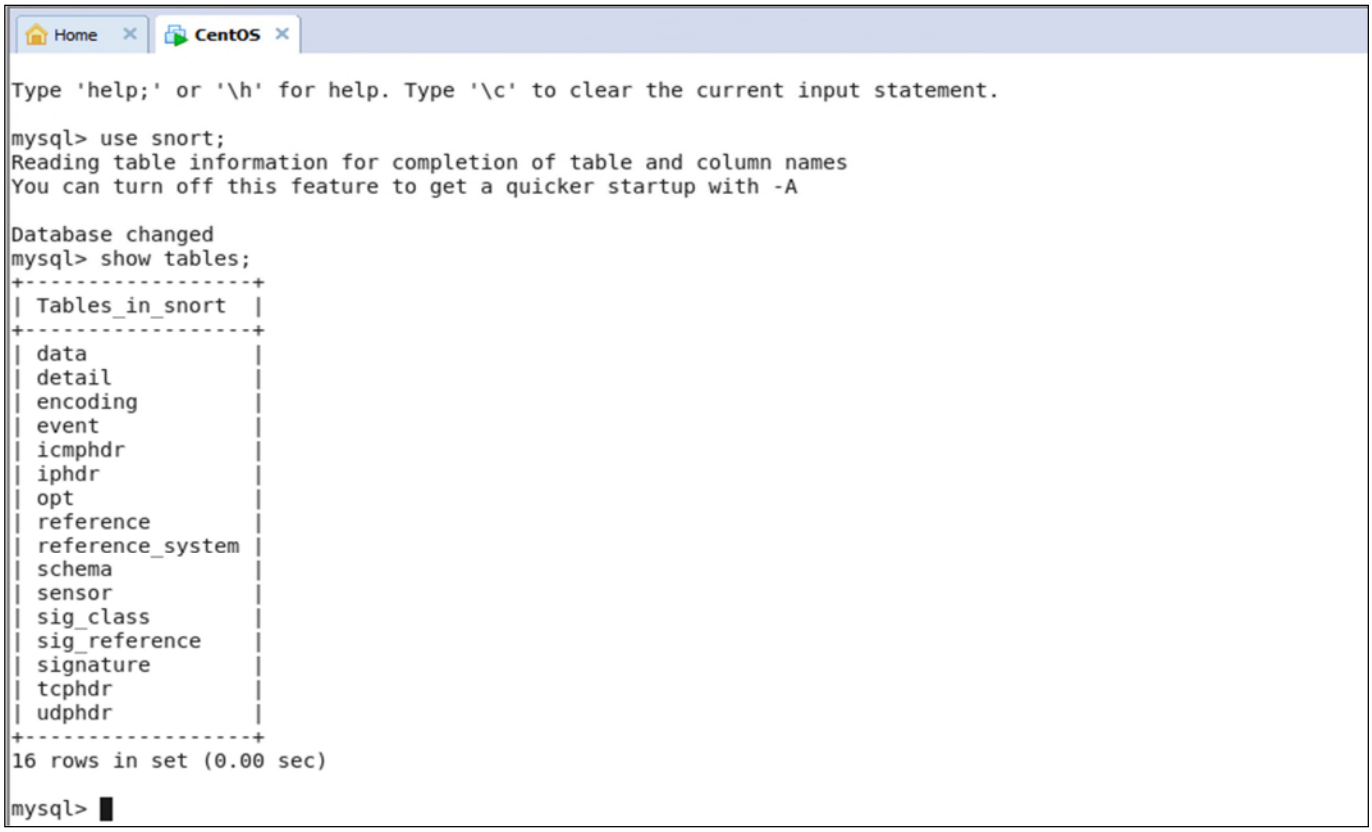



Nous pouvons vérifier si la base SNORT a bien était créer par la commande :
Mysql> show database;




Une fois que nous avons crée les bases de données, 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 :




## V.4. Mise en place de la console B.A.S.E

### V.4.1. Installation des pré-requis

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

### V.4.2. 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 suite

Error_reporting = E_ALL & ~E_NOTICE

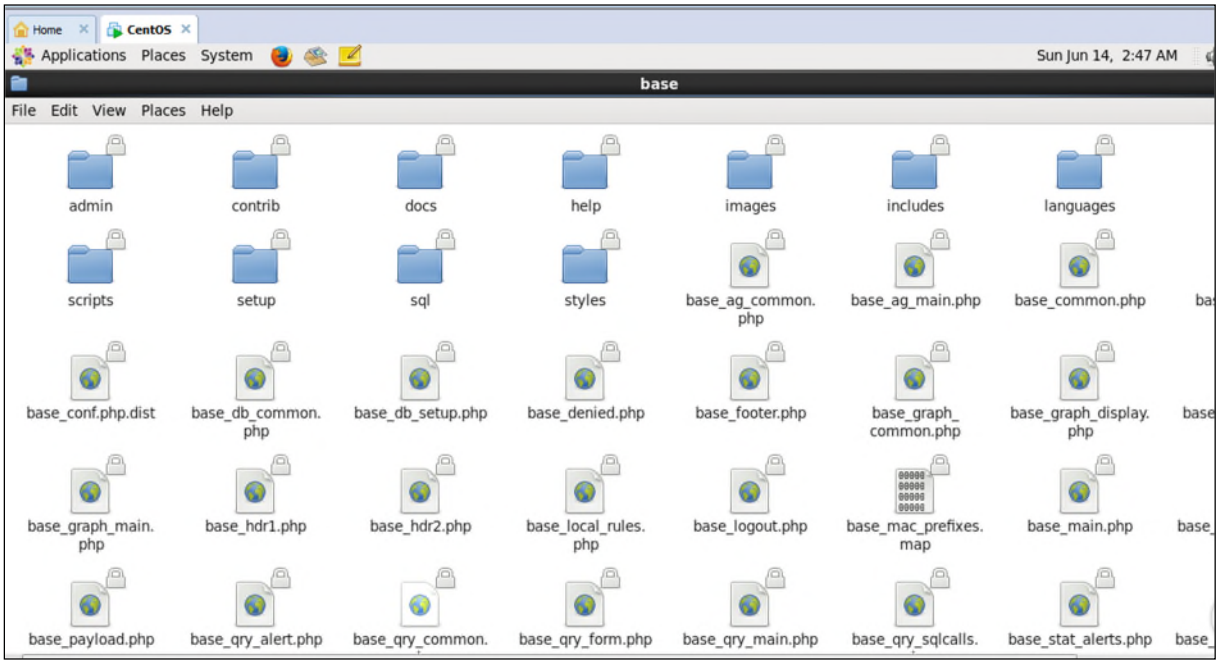
Maintenant nous pouvons démarrer le service http avec la commande
#service httpd start


### V.4.3. Installation de base

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



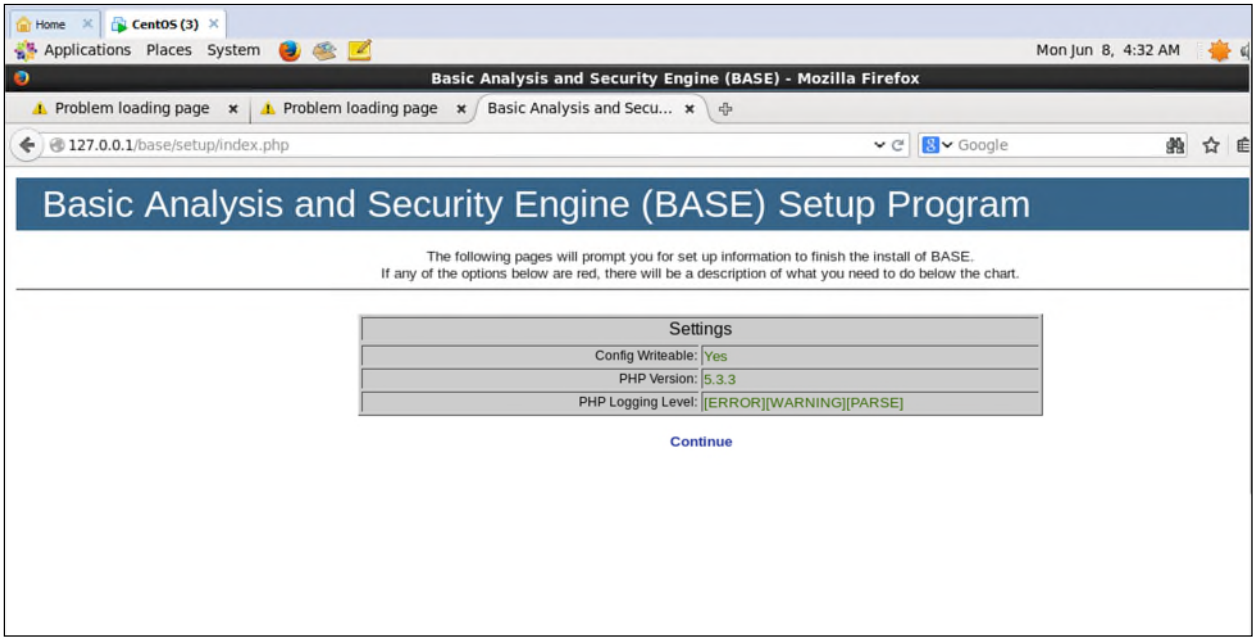
### V.4.4. Installation d'Adodb

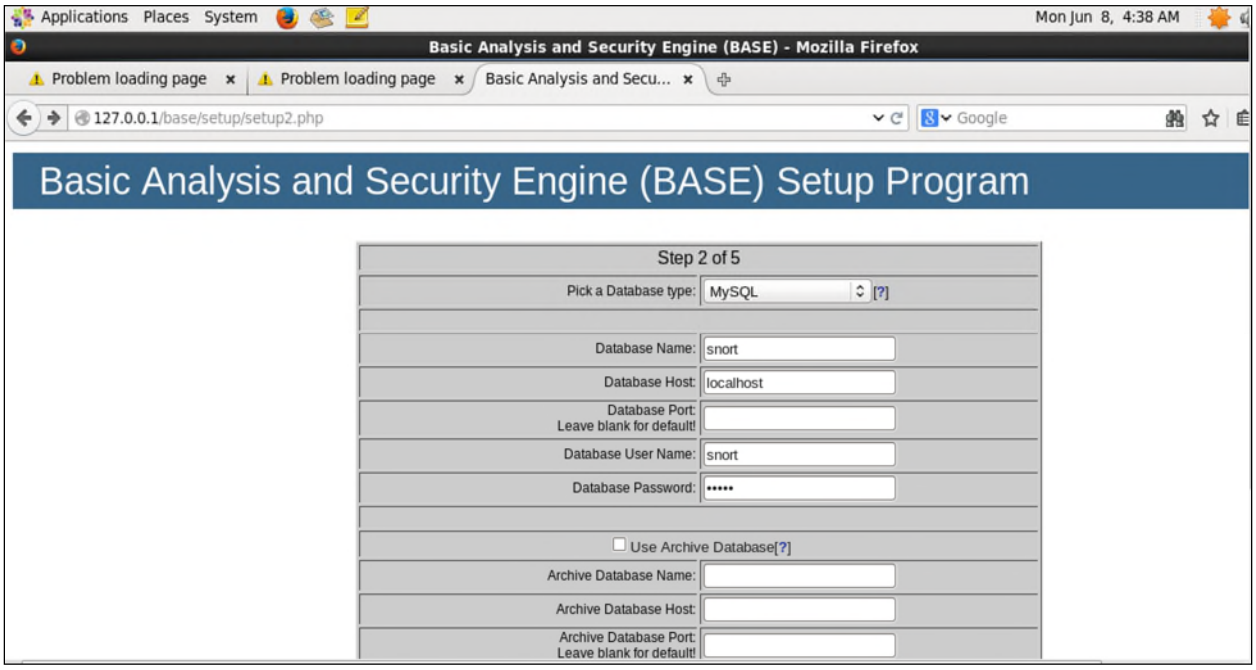
On décompresse le fichier adodb511.tgz avec la commande :
# tar xvzf adodb511.tgz

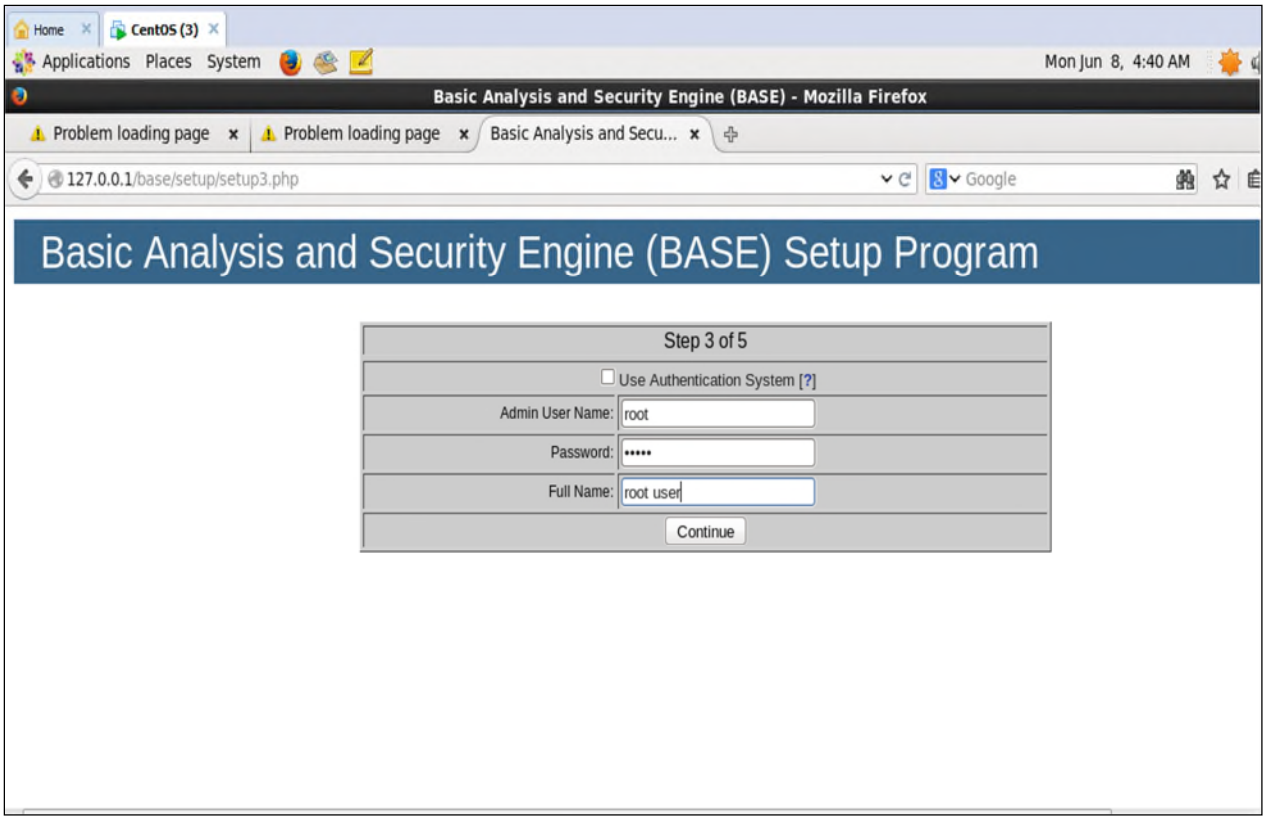
```
[root@localhost /]# ls
adodb511.tgz       bin   etc              Image_Graph-0.8.0.tgz  Log-1.12.8.tgz  mnt          PEAR-1.9.5      pulledpork-0.7.0        sbin
barnyard           boot  home             lib                    lost+found      opt          PEAR-1.9.5.tgz  pulledpork-0.7.0.tar.gz  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
```

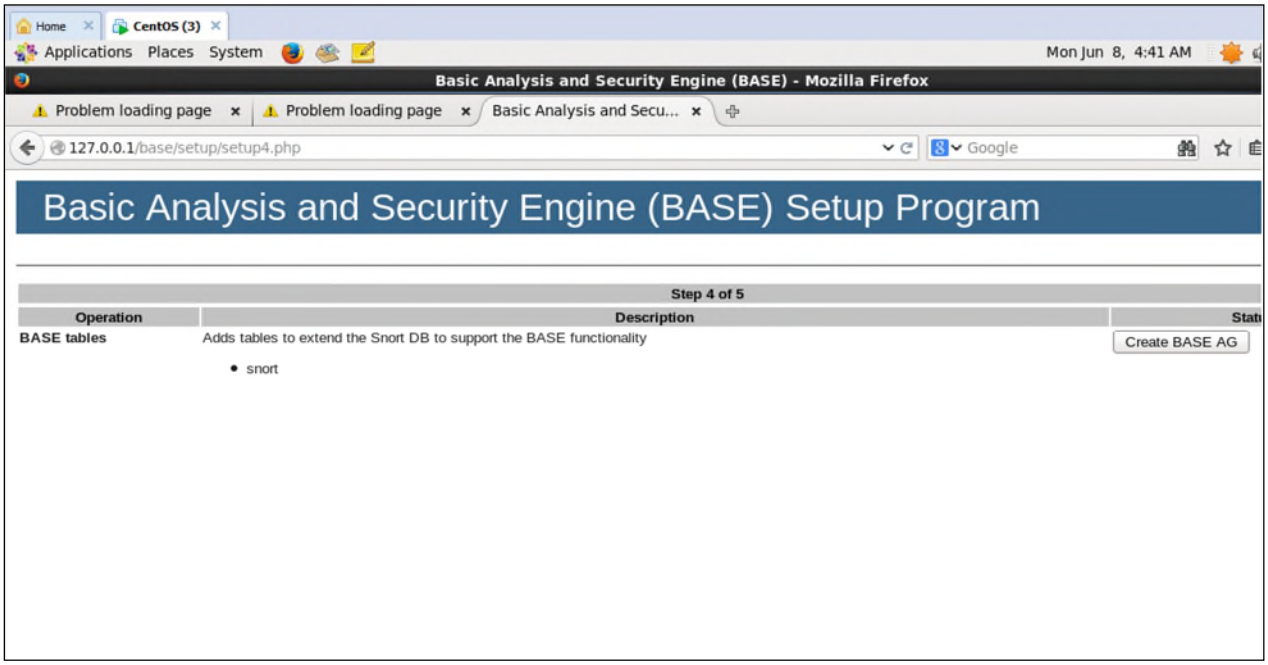
Puis on le déplace dans le répertoire /var/www/base

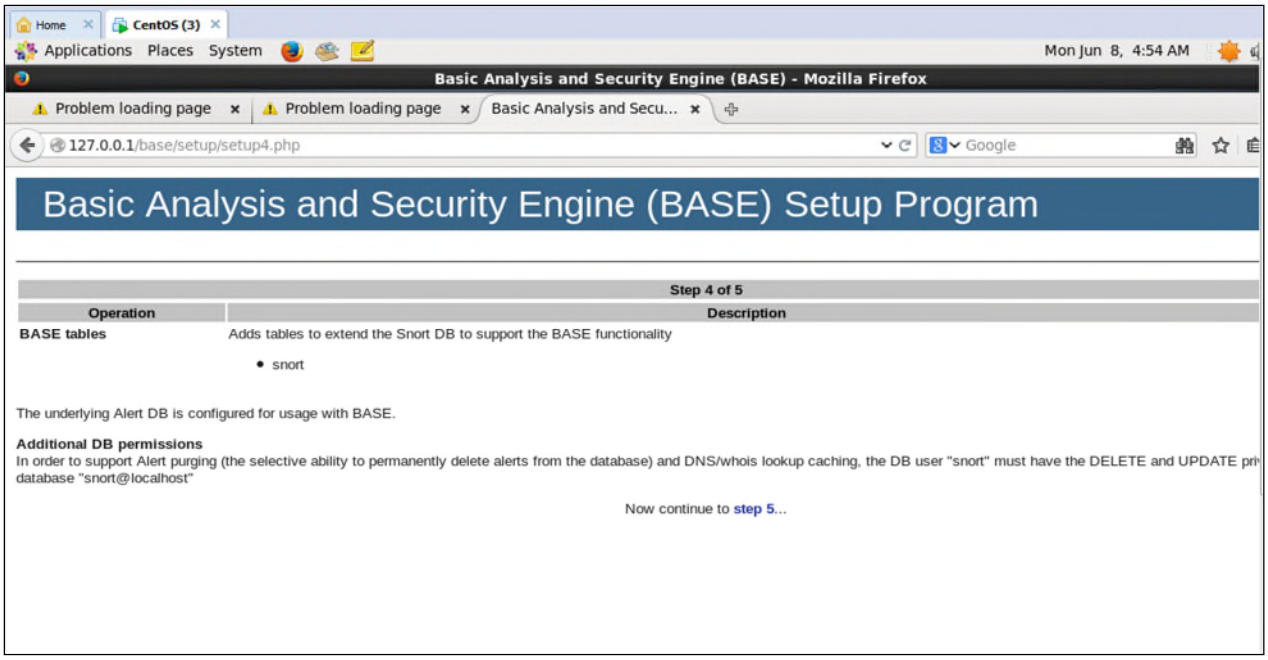
Après l'installation de B.A.S.E, on tape dans le navigateur : 127.0.0.1/base/ on aura l'interface de notre base :

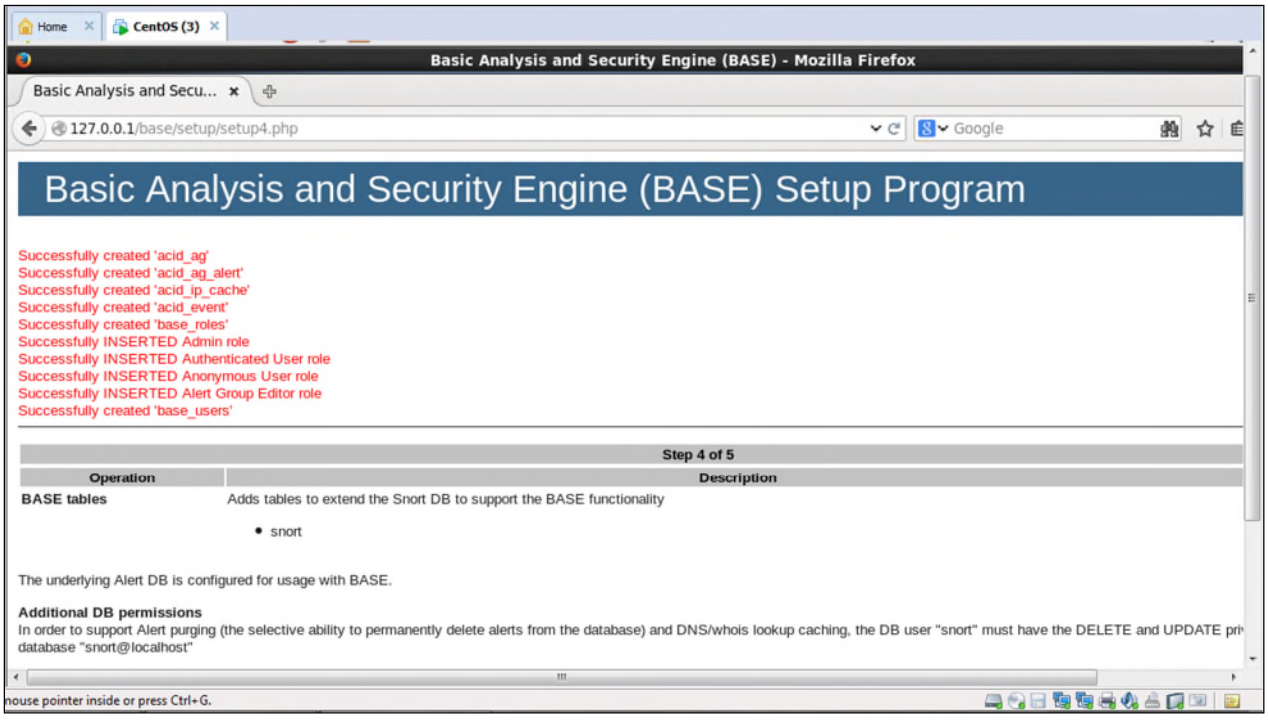


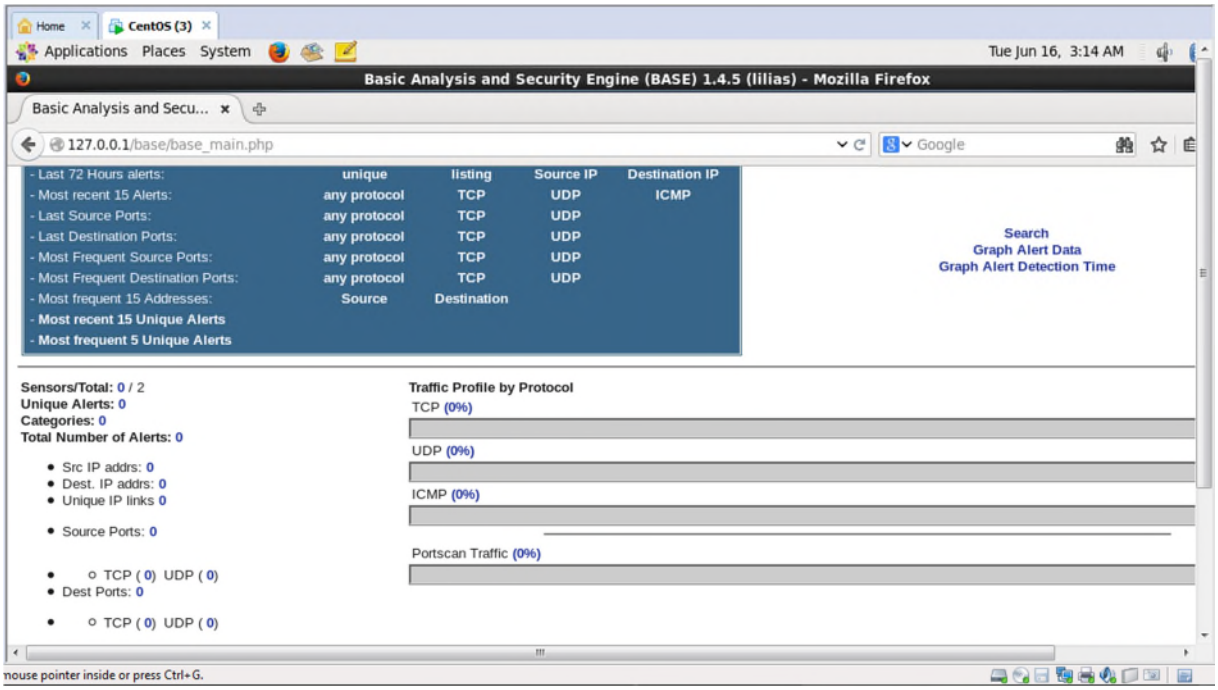




Home
CentOS (3)
Applications Places System
Mon Jun 8, 4:40 AM
Basic Analysis and Security Engine (BASE) - Mozilla Firefox
Problem loading page
Problem loading page
Basic Analysis and Secu...
127.0.0.1/base/setup/setup3.php
Google
Basic Analysis and Security Engine (BASE) Setup Program
Step 3 of 5
Use Authentication System [?]
Admin User Name: root
Password: •••••
Full Name: root user
Continue


Home
CentOS (3)
Applications Places System
Mon Jun 8, 4:41 AM
Basic Analysis and Security Engine (BASE) - Mozilla Firefox
Problem loading page
Problem loading page
Basic Analysis and Secu...
127.0.0.1/base/setup/setup4.php
Google
Basic Analysis and Security Engine (BASE) Setup Program
Step 4 of 5
Operation
Description
BASE tables
Adds tables to extend the Snort DB to support the BASE functionality
Create BASE AG
snort

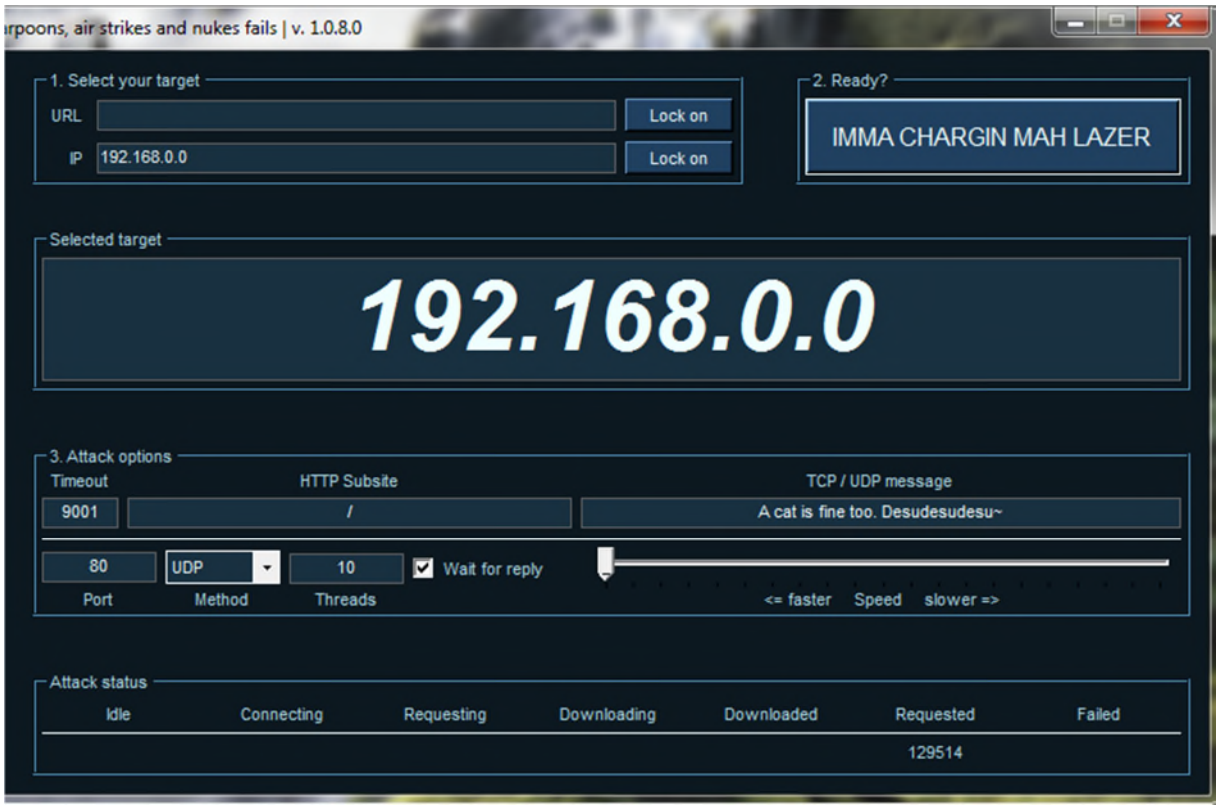

Home
CentOS (3)
Applications Places System
Mon Jun 8, 4:54 AM
Basic Analysis and Security Engine (BASE) - Mozilla Firefox
Problem loading page
Problem loading page
Basic Analysis and Secu...
127.0.0.1/base/setup/setup4.php
Google
Basic Analysis and Security Engine (BASE) Setup Program
Step 4 of 5
Operation
Description
BASE tables
Adds tables to extend the Snort DB to support the BASE functionality
snort
The underlying Alert DB is configured for usage with BASE.
Additional DB permissions
In order to support Alert purging (the selective ability to permanently delete alerts from the database) and DNS/whois lookup caching, the DB user "snort" must have the DELETE and UPDATE pri
database "snort@localhost"
Now continue to step 5...


Home
CentOS (3)
Basic Analysis and Security Engine (BASE) - Mozilla Firefox
Basic Analysis and Secu...
127.0.0.1/base/setup/setup4.php
Google
Basic Analysis and Security Engine (BASE) Setup Program
Successfully created 'acid_ag'
Successfully created 'acid_ag_alert'
Successfully created 'acid_ip_cache'
Successfully created 'acid_event'
Successfully created 'base_roles'
Successfully INSERTED Admin role
Successfully INSERTED Authenticated User role
Successfully INSERTED Anonymous User role
Successfully INSERTED Alert Group Editor role
Successfully created 'base_users'
Step 4 of 5
Operation
Description
BASE tables
Adds tables to extend the Snort DB to support the BASE functionality
snort
The underlying Alert DB is configured for usage with BASE.
Additional DB permissions
In order to support Alert purging (the selective ability to permanently delete alerts from the database) and DNS/whois lookup caching, the DB user "snort" must have the DELETE and UPDATE pri
database "snort@localhost"
nouse pointer inside or press Ctrl+G.

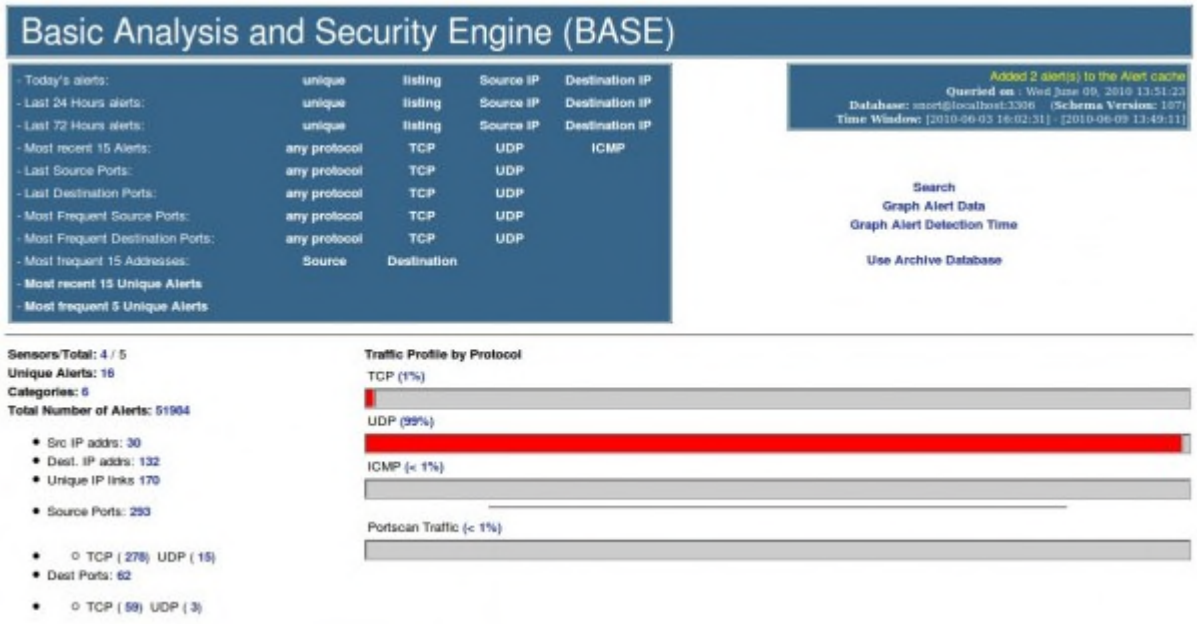



## V.5. lancement d'attaque

Après avoir spécifié l'adresse réseau de l'entreprise (192.168.0.0) dans l'outil d'attaque LOIC et précisé la méthode UDP en cliquant sur le boutton IMMA CHARGING LAZER comme suit :



Nous avons eu le résultat suivant :




## 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, donc gratuit est accessible a n'importe quel utilisateur, il est surtout conseillé aux petites entreprises qui n'ont pas les moyens ou les besoins de procurer des solutions hardwares qui sont très chères, et qui offre le moindre service ou mise a jour avec des tarifs exorbitants.

# Conclusion générale

Il est de nos jours nécessaire de mettre en place un système de détection d'intrusion puisque la mise en place des pare-feux et des systèmes d'authentification ne sont plus suffisants.

Nous avons étudié le fonctionnement de ces IDS, en particulier nous avons pris comme exemple l'outil Open Source Snort qui est le plus réputé en terme d'efficacité et présente une souplesse en terme de personnalisation. Ce qui nous a offert l'occasion de travailler sous l'environnement CentOs, découvrir et enrichir nos connaissances, à savoir les réseaux informatiques, leurs sécurités en générale et les systèmes de détection d'intrusion en particulier.

Grâce à notre étude nous avons aussi constaté qu'on ne peut pas avoir un réseau absolument sécurisé.

# Bibliographie

**[1] Arnould.Gerard,** Etude et Conception d'Architectures Haut-Débit pour la Modulation et la Démodulation Numériques, THÈSE DE DOCTORAT, École Doctorale IAEM – Lorraine Département de Formation Doctorale Électronique – Électrotechnique, Université Paul Verlaine – Metz, Décembre 2006

**[2] Baudoin Karle,** NT réseau IDS et IPS, université Paris-EST Marne-la-vallée par l'enseignant Etienne Duris, 2003/2004.

**[3](David Burgermeister, Jonathan Krier,** « Les systèmes de détection d'intrusions », 2006.

**[4] ESSAIDI, Vivien BOISTUAUD, Ngoné DIOP,** conception d'une zone démilitarisé (DMZ), mémoire de master en informatique, option réseau, université de Marne la vallée. 2006-2007

**[5] Gunadiz Safia,** algorithme d'intelligence artificielle pour la classification d'attaques réseaux à partir de données TCP, Mémoire de magistère, université M'hamed BOUGARA de Boumerdes, 2010/2011

**[6] Guy PUJOLL,** Les réseaux, ouvrage édition EYROLLES, 2008**.**

**[7] Hamza lamia,** Génération automatiques de scénario d'attaques pour les systèmes de détection d'intrusion, Mémoire de magistère, université Abderrahmane Mira de béjaia, 2005.

**[8] Jabou Chaouki, Schillings Michaël, Hantach Anis,** « TER Détection d'anomalies sur le réseau », Rapport de projet, Université Paris. Descartes, 2009.

**[9] Jonathan-CristoferDemay,** Génération et évaluation de mécanismes de détection d'intrusion au niveau applicatif, thèse de doctorat, école doctorale Matisse, université de Rennes 1, Juillet 2011.

**[10] Michael AMAND Mouhamed NSIRI,** Etude d'un système de détection d'intrusion comportemental pour l'analyse du trafic aéroportuaire, Rapport de projet tutoyé, Janvier 2011.

**[11] Mohammed EL-Sayed GADELRAB,** Evaluation des systèmes de détection d'intrusion, Thèse de doctorat, option Systèmes informatiques critiques, université de Toulouse III, 2008.

**[12] Noudjoud KAHYA,** Etude critique des méthodes d'optimisation pour la détection d'intrusion dans un système informatique, mémoire de magistère en informatique, option : Réseaux et systèmes distribués, université Abderrahmane Mira de béjaia, Novembre 2005

**[13] RIAHLA,** Introduction à la sécurité informatique, conférence au département de physique/Infotronique IT/S6 de l'université de Boumerdes, 2008-2009.

**[14] Romdhane ben younes,** Etude et mise en œuvre d'une méthode de détection d'intrusion dans les réseaux sans fil 802.11 basé sur la vérification formelle de modèle, mémoire, université du Québec à Montréal, Décembre 2007**.**

**[15] Thierry Evangelista,** les systèmes de détection d'intrusion informatiques, édition DUNDO, Paris 2004.

**[16] Yann Duchemin,** Apporter les notions essentiels pour l'interconnexion de réseau dans des environnements de communication hétérogène basé sur TCP/IP, avril 2000.

**[17] Eric Farman,** Snort, rapport de stage, département informatique, mairie de Pértuis, France, 2012.

**[18]Fathi Ben Nasr, Alia Khessairi Abbassi,** Mise en place d'une sonde Snort, Mémoire de mastère informatique, spécialité sécurité informatique, Ecole nationale des sciences de l'informatique, 2004-2005.

**[19] Sébastien Durand,** Système de détection d'intrusion Snort, Mémoire de mastère, Institut National des Sciences Appliquées de Rouen, 2002-2003.

# Webographie

**http://www.ingenieurs2000.com**
**http://www.snort.org**
**http://www.securinets.com/sites/default/files/fichiers_pdf/Snort.pdf**
**http://www.mi.parisdescartes.fr/osalem/Projets/CMA.pdf**

# Résumé

Suite à notre étude sur la sécurité dans les réseaux informatiques, on se rend compte que ce n'est pas facile d'assurer la sécurité d'un réseau informatique et de le protéger contre d'éventuelles intrusions.

L'évolution des outils permettant de réaliser des attaques informatiques a mis les différents réseaux des entreprises en danger. Avoir un réseau complètement sécurisé est pratiquement irréalisable. Par conséquent, il est nécessaire de pouvoir détecter les violations de sécurité lorsqu'elles se produisent. Cela est rendu possible grâce aux mécanismes de détection et de prévention des intrusions. La détection d'intrusion consiste à découvrir l'utilisation d'un système informatique à des fins non légales, tandis que les systèmes de prévention des intrusions tentent de les stopper en prévenant le firewall de l'existence d'une tentative d'intrusion.

Snort est un outil utilisé pour détecter les intrusions, plus réputé en termes d'efficacité et présente une souplesse en terme de personnalisation.

# Abstract

Following our study on security in computer networks, one realizes that it is not easy to ensure the security of a computer network and protect against possible intrusions.

The development of tools for performing computer attacks put the networks of enterprises at risk. Having a completely secure network is virtually impossible. Therefore, it is necessary to detect security violations as they occur. This is made possible by the detection mechanisms and intrusion prevention. Intrusion detection is to discover the use of a computer system to non-legal purposes, while intrusion prevention systems attempt to stop preventing the firewall of the existence of an intrusion attempt.

Snort is a tool used to detect intrusions, best known for efficiency and has flexibility in terms of customization.