Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de licence en Informatique

Option : informatique générale

Thème

Etude et mise en place d'un système de détection/prévention d'intrusion (IDS/IPS) réseau. Etude de cas SNORT

Réalisé par :

- DABOUR Imane
- HADJI Imène
-

Présenté le .. Juin 2014 devant le jury composé de MM.

- *Mr BENAISSA Mohammed Samir* *(Encadreur)*
- *Mr.Tadlaoui.M* *(Examinateur)*
- *Mme.Iles.N* *(Examinateur)*

Année universitaire : 2013-2014

Remerciements

En préambule à ce mémoire nous remercions ***ALLAH*** qui nous aide et nous donne la patience et le courage durant ces années d'étude.

Nous souhaitons d'adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

Ces remerciements vont tout d'abord au corps professoral et administratif de département d'informatique de l'université ABOU BAKR BELKAID de Tlemcen pour la richesse et la qualité de leurs enseignements et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Ensuite nous tenons à remercier notre encadreur ***Mr BENAISSA Mohammed Samir***pour l'orientation, la confiance, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Nous tenons aussi à remercier les membres du jury qui ont accepté d'examiner notre mémoire.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenue et encouragé au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Dédicace

Je dédié ce modeste travail :

À la plus belle créature que Dieu a créée sur terre,,,

À cet source de tendresse, de patience et de générosité,,,

À ma mère ***MOSTEFAI Fatima***

À ma sœur ***Nesrine*** *qui a toujours était à mes cotés*

*À*Mes oncles, mes tantes, et à toute ma famille.

*À*mon binôme **Imène**, et à tous Mes amis.

Je remercie également tous mes professeurs et surtout mon encadreur

Mr BENAISSA Mohammed Samir

En un mot à tous les gens qui contribué ma réussite de près ou de loin.

DABOUR Imane

Dédicace

A l'aide de ***DIEU*** tout puissant, qui trace le chemin de ma vie, j'ai pu arriver à réaliser ce

Modeste travail que je dédie :

A la mémoire de ma grande mère paternel « **Belaidi Zahra** » pour toutes ses prières.

A ma très chère mère « HADJOU BELAID Souad »

*A*ffable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'as pas cessé de m'encourager et de prier pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance , durant mon enfance et même à l'âge adulte . Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.

Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A ma plus belle étoile ***« DABOUR Imane »*** et sa mère : ma tente ***« Fatima »***qui puisse exister dans l'univers, que je les souhaite une longue vie pleine de joie, de réussite, de bonheur, et de santé.

A mon petit frère ***« YOUCEF »*** que je leur souhaite une longue vie pleine de joie et de réussite dans sa vie et ses études.

Dédicace

A mes sœurs adorables *« **RANIA, KHAWLA***, et ma petite sœur ***SALSABILE***».

A mes enseignants et surtout mon professeur, mon encadreur, mon père « **benaissamohammed** »qui ma aidé tout au long de ce mémoire et que je leur souhaite une très belle vie plein de joie, de santé, et de bonheur.

Et à tout ceux qui m'aiment et qui me cannaient de proche ou de loin.**HADJI Imène**

Chapitre 3 : Outil de détection intrusion Snort

Introduction générale

Les réseaux informatiques sont devenus des ressources vitales et déterministes pour le bon fonctionnement des entreprises. De plus, ces réseaux sont ouverts de fait qu'ils sont pour la plus part raccordés à l'Internet.

Cette ouverture qui permet de faciliter la communication, engendre malheureusement des risques importants dans le domaine de la sécurité informatique. Les utilisateurs de l'Internet ne sont pas forcements pleins de bonnes intentions, ils peuvent exploiter les vulnérabilités des réseaux et systèmes pour réaliser leurs attaques. Les conséquences de ces attaques peuvent être lourdes pour un particulier (pertes d'informations, ou pire encore vol d'informations, atteinte à la vie privée..) et pour une entreprise (perte du savoir-faire, atteinte à l'image de marque, perte financière..). Pour cela, les administrateurs déploient des solutions de sécurité efficace capable de protéger le réseau de l'entreprise. Dans ce contexte, les IDS constituent une bonne alternative pour mieux protéger le réseau informatique.

Un système de détection d'intrusion (IDS) est un mécanisme écoutant le trafic réseau de manière furtive afin de repérer des activités anormales ou suspectes et permettant aussi d'avoir une action de prévention sur les risques d'intrusion.

Dans le cadre de ce projet nous nous intéresserons aux outils de détection d'intrusions réseaux (IDS) plus particulièrement à snort , permettant de détecter des intrusions réseau à temps réel .

L'objectif de notre travail est :

- Etudier et analyser toute les aspects traités par un système de détection / prévention d'intrusion réseau.
- Etude de cas : snort
- Installation et configuration de snort
- Tests de détection d'intrusion en utilisant :
- des règles prédéfinies de snort.
- Tests et évaluations de performance de IDS Snort.

Ce mémoire va être organisé comme suit :

Dans le premier chapitre, nous présentons l'étude des systèmes détections intrusion : IDS

Dans le second chapitre, nous allons détailler des généralités sur les attaques réseaux.

Enfin, le dernier chapitre, Nous expliquerons le fonctionnement de l'outil de détection intrusion snort.

1.1. Introduction

Un système de détection d'intrusion (ou IDS : Intrusion Detection System) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une action de prévention et d'intervention sur les risques d'intrusion.

Afin de détecter les attaques que peut subir un système (réseau informatique), il est nécessaire d'avoir un logiciel spécialisé dont le rôle est de surveiller les données qui transitent sur ce système, et qui est capable de réagir si des données semblent suspectes.

1.2. Principes de Fonctionnement des IDS

1.2.1. Méthodes de Détection des IDS

Pour bien gérer un système de détection d'intrusions, il est important de comprendre comment celui-ci fonctionne :

- Comment reconnaître/définir une intrusion?
- Comment une intrusion est-elle détectée par un tel système ?
- Quels critères différencient un flux contenant une attaque d'un flux normal ?

Ces questions nous ont amené à étudier le fonctionnement interne des IDS .

Il existe plusieurs méthodes permettant de détecter une intrusion :

➔ La première consiste à détecter des signatures d'attaques connues dans les paquets circulant sur le réseau : **l'Approche par scénario ou par signature.**

➔ La seconde, consiste quant à elle, à détecter une activité suspecte dans le Comportement de l'utilisateur : **l'Approche comportementale ou par Anomalie.**

Ces de9ux techniques, aussi différentes soient-elles, peuvent être combinées au sein d'un même système afin d'accroître la sécurité [2].

1.2.1.1. Approche par scénario ou par signature

Cette technique s'appuie sur la connaissance des techniques utilisées par les attaquants pour déduire des scénarios typiques.

Elle ne tient pas compte des actions passées de l'utilisateur et utilise des signatures d'attaques existantes (ensemble de caractéristiques permettant d'identifier une activité intrusive : une chaîne alphanumérique, une taille de paquet inhabituelle, une trame formatée de manière suspecte, …) .

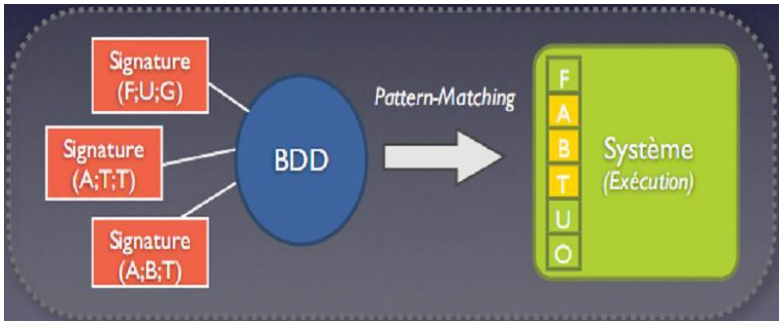

Figure 1.1. : Approche par scénario ou par signature

Cette technique se base sur :

- **La recherche de motifs (pattern matching) :**

C'est la méthode la plus connue et la plus facile à comprendre.

Elle se base sur la recherche de motifs (chaînes de caractères ou suite d'octets) au sein du flux de données.

L'IDS comporte une base de signatures où chaque signature contient les protocoles et ports utilisés par une attaque spécifique ainsi que le motif qui permettra de reconnaître les paquets suspects.

De manière analogue, cette technique est également utilisée dans les anti-virus. En effet un anti-virus ne peut reconnaître un virus que si ce dernier est reconnu dans sa base de signatures virale, d'où la mise à jour régulière des anti-virus.

- **Recherche de motifs dynamiques**

Le principe de cette méthode est le même que précédemment mais les signatures des attaques

Évoluent dynamiquement. L'IDS est de ce fait doté de fonctionnalités d'adaptation et d'apprentissage.

- **Analyse de protocoles**

Cette méthode se base sur une vérification de la conformitédes flux,ainsi que sur l'observation des champs et paramètres suspects dans les paquets. L'analyse protocolaire est souvent implémentée par un ensemble de préprocesseurs (programmes ou plug-in), où chaque préprocesseur est chargé d'analyser un protocole particulier (FTP, HTTP, ICMP, ...). Du fait de la présence de tous ces préprocesseurs, les performances dans un tel système s'en voient fortement dégradées (occupation du processeur).
L'intérêt fort de l'analyse protocolaire est qu'elle permet de détecter des attaques inconnues, contrairement au pattern matching qui doit connaître l'attaque pour pouvoir la détecter.

- **Analyse heuristique et détection d'anomalies**

Le but de cette méthode est, par une analyse intelligente, de détecter une activité suspecte ou toute autre anomalie (une action qui viole la politique de sécurité définie dans l' IDS) .
Par exemple : une analyse heuristique permet de générer une alarme quand le nombre de pings vers un réseau ou hôte est très élevé ou incessant (Ping de la mort).

1.2.1.2. L'approche comportementale (AnomalyDetection)

Cette technique consiste à détecter une intrusion en fonction du comportement passé de l'utilisateur. Il faut préalablement dresser un profil utilisateur à partir de ses habitudes et déclencher une alerte lorsque des événements hors profil se produisent.
Cette technique peut être appliquée non seulement à des utilisateurs mais aussi à des applications et services.

Plusieurs métriques (paramètres) sont possibles : la charge CPU, le volume de données échangées, le temps de connexion sur des ressources, la répartition statistique des protocoles et applications utilisés, les heures de connexion, ...

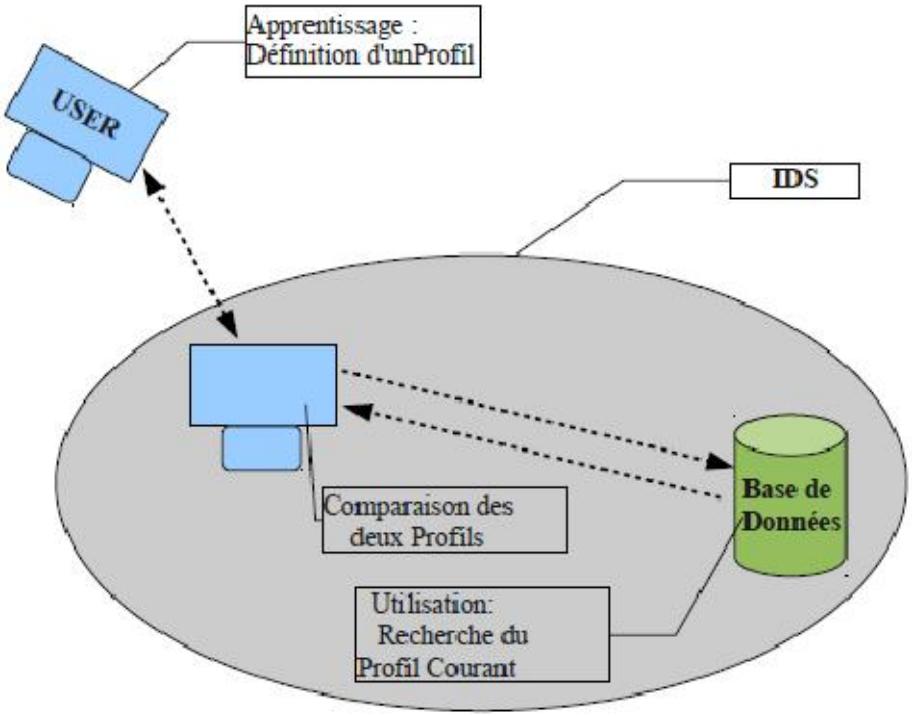

Figure 1.2 : Illustration de l'approche comportementale

1.3. Architecture des IDS

Un IDS est essentiellement constitué d'un sniffer couplé avec un moteur qui analyse le trafic et entreprend des actions suivants les règles définies dans l'IDS. Ces règles décrivent le comportement de l' IDS selon le trafic analysé trafic :

Alertes, journalisation des événements dans des fichiers logs.

Un IDS peut analyser les couches suivantes :

- Couche Réseau (IP, ICMP)
- Couche Transport (TCP, UDP)

- Couche Application (HTTP, Telnet)

Selon le type de trafic, l' IDS accomplit certaines actions définies dans les règles . Certains termes sont souvent employés quand on parle d' IDS :

- **Faux positif** : une alerte provenant d'un IDS mais qui ne correspond pas à une attaque réelle (Fausse Alerte) .

- **Faux négatif** : une intrusion réelle qui n'a pas été détectée par l'IDS

Le schéma suivant illustre le fonctionnement et les caractéristiques d'un IDS :

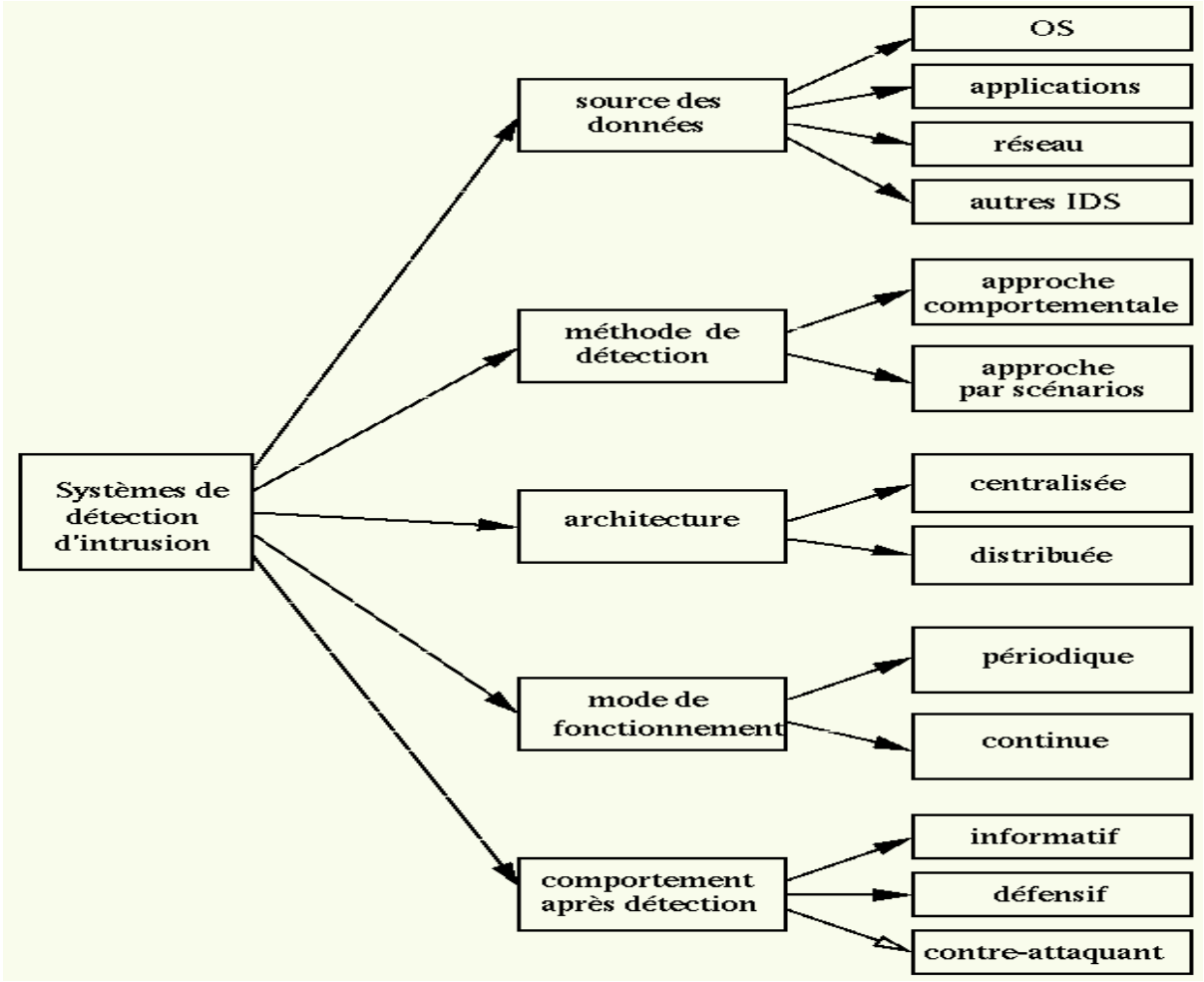

Figure 1.3 : Caractéristiques et Fonctionnement des IDS

1.4. Différents Types IDS

Il existe plusieurs types d'IDS, mais on peut les classer en deux familles :

- **Les NIDS** : Network IDS, système de détection d'intrusion réseau
- **Les HIDS** : Host IDS, système de détection d'intrusion de type hôte

Les autres IDS sont en réalité des dérivées de ces familles : les IDS Hybrides, les IPS (systèmes de prévention d'intrusions).

Les IDS sont disponibles sous formats :

➔ **Les logiciels** : permettent à n'importe quel administrateur de réseau de l'installer sur sonOS. Ils sont faciles à installer, configurer et contrôler. Cependant, cet OS (Windows, Linux) est une distribution dont les failles peuvent être connues des hackers. Il est plus vulnérable si les patchs (programmes de mise à jour) ne sont pas régulièrement installés et que les modules inutilisés de l'OS sont conservés.

Les NIDS

Les NIDS sont des IDS dédiés aux réseaux. Ils comportent généralement une sonde (machine par exemple) qui "écoute" sur le segment de réseau à surveiller , un capteur et un moteur qui réalise l'analyse du trafic afin de détecter les intrusions en temps réel. Un NIDS écoute donc tout le trafic réseau, puis l'analyse et génère des alertes si des paquets semblent dangereux.

Les IPS

Les IPS ont pour fonction principale d'empêcher toute activité suspecte détectée au sein d'un système : ils sont capables de prévenir une attaque avant qu'elle atteigne sa destination.
Contrairement aux IDS, les IPS sont des outils aux fonctions « actives », qui en plus de détecter une intrusion, tentent de la bloquer.
Le principe de fonctionnement d'un IPS est analogue à celui d'un IDS, ajoutant à cela l'analyse des contextes de connexion, l'automatisation d'analyse des logs et la coupure des connexions suspectes. Contrairement aux IDS classiques, aucune signature n'est utilisée pour

détecter les attaques. Avant toute action, une décision en temps réel est exécutée (i.e., l'activité est comparée aux règles existantes). Si l'action est conforme à l'ensemble de règles, la permission de l'exécuter sera accordée et l'action sera exécutée. Si l'action est illégale (c'est-à-dire si le programme demande des données ou veut les changer alors que cette action ne lui est pas permise), une alarme est générée. Dans la plupart des cas, les autres détecteurs du réseau (ou une console centrale connectées à l'IPS) en seront aussi informés dans le but d'empêcher les autres ordinateurs d'ouvrir ou d'exécuter des fichiers spécifiques (si on est en réseau) .

On peut classer les IPS en deux groupes suivants leurs domaines d'utilisation :

- **Les NIPS** : Network IPS dédiés aux réseaux. Ils ont les mêmes fonctions que les IDS Classiques, sauf qu'ils ont la capacité d'anticiper une attaque.
- **Les HIPS/KIPS** : Host IPS, plus connu sous l'appellation de systèmes de prévention d'intrusions « kernel » (**KIPS**) , spécifiques aux hôtes . Ils supervisent l'intégralité des activités sur la machine où elle est déployée.

L'utilisation d'un détecteur d'intrusions au niveau noyau peut s'avérer parfois nécessaire pour sécuriser une station.

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 c'est une solution rarement utilisée sur des serveurs souvent sollicités.

Exemple de KIPS : Secure IIS, qui est une surcouche du serveur IIS de Microsoft.

Cependant les IPS possèdent quelques inconvénients: ils bloquent toute activité qui semble suspecte.

- Les **firewalls** ne sont pas des IDS à proprement parler mais ils permettent également de stopper des attaques. Les firewalls sont basés sur des règles statiques afin de contrôler les flux entrant et sortant. Ils travaillent en général au niveau des couches basses du modèle OSI (jusqu'au niveau 4), ce qui est insuffisant pour stopper une intrusion.

.

Figure 1.4 : Les firewalls

1.5. Critères de Choix D'un IDS

Les systèmes de détection d'intrusion sont devenus indispensables lors de la mise en place d'une infrastructure de sécurité opérationnelle. Ils s'intègrent donc toujours dans un contexte et dans une architecture imposants des contraintes très diverses.

Certains critères imposant le choix d'un IDS peuvent être dégagés:

- **Fiabilité** : Les alertes générées doivent être justifiées et aucune intrusion ne doit pouvoir lui échapper.

- **Réactivité** : Un IDS doit être capable de détecter les nouveaux types d'attaques le plus rapidement possible ; pour cela il doit rester constamment à jour. Des capacités de mise à jour automatique sont indispensables.

- **Facilité de mise en œuvre et adaptabilité** : Un IDS doit être facile à mettre en œuvre , surtout s'adapter au contexte dans lequel il doit opérer .

Il est inutile d'avoir un IDS émettant des alertes en moins de 10 secondes si les ressources nécessaires à une réaction ne sont pas disponibles pour agir dans les mêmes contraintes de temps.

- **Performance** : la mise en place d'un IDS ne doit en aucun cas affecter les performances des systèmes surveillés.

De plus, il faut toujours avoir la certitude que l'IDS a la capacité de traiter toute l'information à sa disposition (par exemple un IDS réseau doit être capable de traiter l'ensemble du flux pouvant se présenter à un instant donné sans jamais supprimer de paquets) car dans le cas contraire il devient trivial de masquer les attaques en augmentant la quantité d'information.

1.6. Choix du placement d'un IDS

Le placement des IDS va dépendre de la politique de sécurité définie dans le réseau. Mais il existe des positions qu'on peut qualifier de standards, par exemple il serait intéressant de placer des IDS :

- Dans la zone démilitarisée (attaques contre les systèmes publics) **.**

- Dans le (ou les) réseau(x) privé(s) (intrusions vers ou depuis le réseau interne) .
- Sur la patte extérieure du firewall (détection de signes d'attaques parmi tout le traficentrant et sortant, avant que n'importe quelle protection intervienne).

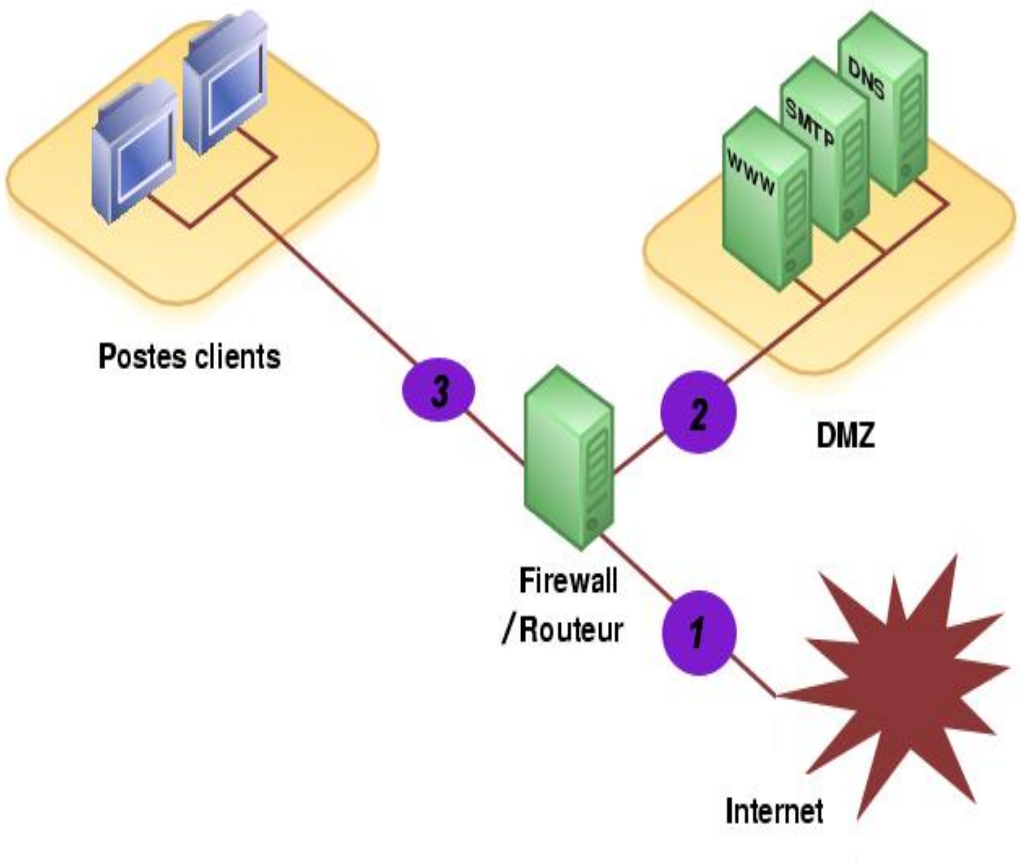

Figure 1.5 : Choix du Placement d'un IDS

Il est important de bien définir les zones sensibles du système (réseau), ainsi que les zones les plus attractives pour un pirate. Il faut aussi voir qu'au-delà de l'architecture du réseau, il faut prendre en compte l'organisation de la sécurité existante:

- Recherche-t-on une administration centralisée ?
- Quel est l'existant organisationnel de la surveillance du réseau ?
- Quels sont les compétences et les moyens en internes pour gérer les IDS ?

1.7. Quelques exemples IDS

Face aux menaces d'intrusions, il existe plusieurs solutions concernant le choix d'unIDS. Il existe des solutions commerciales aussi bien qu' ' Open Source. Les solutions Open Source N'ont rien n'à envier aux solutions commerciales. Mieux les solutions commerciales se basent même sur les Open Source pour améliorer leur produit.

La différence notoire entre ces deux solutions se trouve essentiellement sur le déploiement (éventuellement sur le prix!).

Elle nécessite beaucoup de prés-requis telles que des utilitaires de base ou encore des connaissances sur le système où le produit va être déployé. Cette situation se présente surtout quand on est dans un environnement Linux ! Dans l'environnement Windows on ne fait que suivre les instructions du produit en cochant/décochant des cases, faisant des «suivant».

Pour les solutions commerciales nous avons entre autres :

- **Symantec – Symantec Client Security**

Symantec Client Security fournit la protection des clients contre des menaces complexes sur l'Internet en intégrant l'antivirus, le par feu et la détection des intrusions, à travers la gestion et la réponse centralisées. Il aide à protéger l'entreprise contre les virus, les pirates et les menaces combinées.

Cette nouvelle solution fournit un déploiement commun et une fonction de mise à jour pour des technologies de sécurité multiples, permettant une sécurité plus complète du client. SymantecTM Client Security est une solution facile à administrer qui garantit une sécurité multi couches performante.

En protégeant le réseau de l'entreprise avec Symantec, on bénéficie d'une protection constamment à jour contre les virus, les pirates, les intrusions et les menaces combinées. Les technologies de pointe de détection d'intrusion et de protection de pare-feu masquent automatiquement les postes de travail et bloquent les connexions suspectes.

Elles interagissent également en toute transparence avec Symantec AntiVirus pour protéger les postes de travail, serveurs de fichiers et ordinateurs distants contre les virus, les vers, les chevaux de Troie et les menaces combinées.

Les outils d'administration centralisée offrent une protection automatique en temps réel et facilitent la mise à jour de la sécurité du réseau à partir d'un seul emplacement.

Toujours dans les solutions commerciales on peut citer aussi : CSA CISCO, McAfee-Entercept, ISS RealSeacure …

Pour les solutions open source, il y' a une diversité fonctionnant aussi bien sous Windows que sur Linux,quelques unes parmi d'autres :

- **Nessus**

Nessus est un scanneur de vulnérabilités. Avec un outil comme Nessus, il est possible de Scanner le réseau pour tester des failles connues sur l'ensemble du réseau à la fois, sur une ou plusieurs machines, cela est paramétrable.

Couplé à un véritable scanner de ports comme Nmap, il devient possible de tester tous les ports de chaque machine afin de trouver des erreurs de configuration ou de déceler si des services tournent sur des machines alors qu'ils ne devraient pas.

Nmap est un outil très puissant qui donne la possibilité de faire des scans de ports furtifs permettant de passer inaperçu aux yeux des IDS.

On se sert de ce type d'outil afin de jouer au hacker! En effet, il est préférable d'utiliser les mêmes outils que les hackers sur notre réseau afin de voir par nous-mêmes les failles auxquelles nous pourrions être sensibles plutôt que d'attendre que quelqu'un de malveillant transperce nos défenses. Nessus est livré avec une grande panoplie d'attaques .Exemple : attaques par force brute. On peut aussi aisément ajouter d'autres scanneurs de ports ou le coupler avec des outils de force brute, qui couplé avec des dictionnaires bien choisis , permettra de tester les mots de passe employés dans différents services. Cela permet de tester si un mot de passe est capable de résister au minimum requis .Donc en plus de la fonction d ' IDS Nessus peut être un outil d 'audit .

- **Snort :** c'est un IDS open source . Il est capable d'analyser le trafic sur le réseau en temps réel et les paquets circulant sur le réseau. Il concurrence actuellement encore plusieurs produits commerciaux et il y a même certains produits qui se basent sur ce programme ou son moteur de recherche afin de construire leur solution par-dessus.

Il peut exécuter l'analyse de protocole, et peut être employé pour détecter une variété d'attaques, des tentatives comme des débordements, des balayages de port de dérobée ...

Snort emploie un langage flexible de règles, aussi bien qu'un moteur de détection qui utilise une architecture plug-in modulaire. Snort a des possibilités en temps réel d'alerter.
Snort a trois utilisations primaires. Il peut être employé en tant qu'un renifleur de paquets (comme tcpdump), un enregistreur de paquet ou comme plein système de détection d'intrusion réseau.

1.8. Limite Des IDS

Comme tout système informatique, les IDS ont des limites. On peut en citer :

- **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 partiel ou total …
- **Consommation de ressources** : outre la taille des fichiers de logs (de l'ordre du Go), 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 .
- **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.

- **Vulnérabilité aux dénis de service** : un attaquant peut essayer de provoquer un déni de service au niveau du système de détection d'intrusion, ou pire au niveau du système d'exploitation de la machine supportant l' IDS.

Une fois l'IDS désactivé (« hors service »), l'attaquant peut tenter tout ce qui lui convient.

1.9. Conclusion

Dans ce chapitre, nous avons présentés les généralités sur les systèmes de détection des intrusions IDS.

Le système de détection intrusion est un moyen de sécurité et de protection important dans un réseau informatique.

Dans le deuxième chapitre, nous sommes intéressés par les différentesattaques Réseaux.

2.1. Introduction

Des questions importantes se posent quand on traite l'aspect de la sécurité réseau : pourquoi sécuriser un réseau ? Contre qui ? Et comment ?

Un réseau informatique est un ensemble d'équipements interconnectés en vue de transmettre un signal porteur d'une information . Il permet aussi de partager des ressources (fichiers,imprimantes, ...) et offre des services (http,ftp,dns ...) . Dés lors un problème se pose par rapport à l'accès aux ressources et services : les droits d'accès des utilisateurs , des machines internes ou externes (du réseau) . N'importe qui ne doit pas faire n'importe quoi quand il est dans le réseau , surtout quand il n'est pas autorisé à accéder au réseau .

D'ou la nécessité de sécuriser le réseau , contre tout ce qui constitue une menace .

2.2. La sécurité réseau

La sécurité d'un réseau informatique peut être définie sur trois niveaux :

- **Niveau physique**
- **Niveau logique**
- **Niveau réseau**

2.2.1. Sécurité au niveau physique

La sécurité au niveau physique concerne essentiellement l'accès aux bâtiments et aux équipements qu'ils abritent. À ce niveau l'accès même aux bâtiments doit être défini pour savoir qui y a accès ou non . Car un pirate (ou une personne mal intentionnée) peut avoir besoin d'accéder aux locaux du réseau avant de préparer son attaque depuis l'extérieur.

2.2.2. Sécurité au niveau logique

Au niveau logique , la notion de sécurité est plus sensible , les utilisateurs sont en contact direct ou indirect avec le système (données , logiciels , applications , services...) . L'accès au système par tout le monde à un instant voulu peut engendrer des risques qui pourraient exposer ce dernier . Les utilisateurs doivent être « casés » dans des groupes dont chacun a un niveau d'accès bien défini par l'administrateur réseau , suivant les besoins du groupe. Si tout le monde a les mains libres pour accéder au système , une maladresse (erreur) pourrait provoquer des effets très graves et exposer le système à des attaques externes (si le réseau est interconnecté avec l'extérieur par exemple) . En plus seuls les services et les applications nécessaires doivent être installés (ou activés) dans les postes des utilisateurs . Une solidité voire une complexité des mots de passe est également recommandée , car à partir des postes ou comptes clients , un intrus pourrait accéder aux serveurs suivant le niveau d'accès (c'est le cas d'une attaque par rebond) .

Et mieux encore si le réseau local a un accès vers Internet , alors l'accès au web doit être très restreint : n'importe quel site ne doit être autorisé , n'importe quel fichier ne doit pas être téléchargé depuis Internet .

Encore moins n'importe quelle application ne doit être installée sans l ' avis de l'administrateur réseau ou système . Sinon c'est l'intégrité du système qui est exposée à toute menace extérieure. Donc les utilisateurs constituent un maillon essentiel et sensible de la sécurité du réseau.

2.2.3. Sécurité au niveau réseau

Un autre volet très important de la sécurité réseau est l'accès au réseau. Il relève beaucoup d'interrogations :

Qui a accès au réseau ? Quel est son niveau d'accès (droits sur les fichiers et répertoires, privilèges sur les services …) une fois connecté au réseau ? Que faire pour limiter l'accès à certaines ressources du réseau (serveurs par exemple) ?

Il serait très dangereux voire inimaginable d'autoriser l'accès aux ressources internes du réseau à tout le public (quel qu'il soit) . Il existe plusieurs mécanismes pour répondre à ces questions :

- Mise en place de serveur d'authentification (exemple serveur radius) pour identifier

Tout utilisateur voulant se connecter. Il existe deux niveaux d'authentification : authentification simple et forte.

L'authentification simple se base sur l'usage de login et de mots de passe. Alors que l'authentification forte se base sur les certificats électroniques, les cartes à puce, l'index du doigt … avec la création de tunnels sécurisés lors de l'authentification.

- Mise en place de Firewall. Les firewalls qui permettent de filtrer l'accès au réseau.

- Mise en place de DMZ (zone démilitarisée) . Une DMZ permet de partager certaines

Ressources du réseau local avec l'extérieur. Cette zone est accessible depuis l'intérieur comme l'extérieur du réseau.

Le schéma ci-dessous rappelle très sommairement les différents niveaux pour lesquels un risque en matière de sécurité existe :

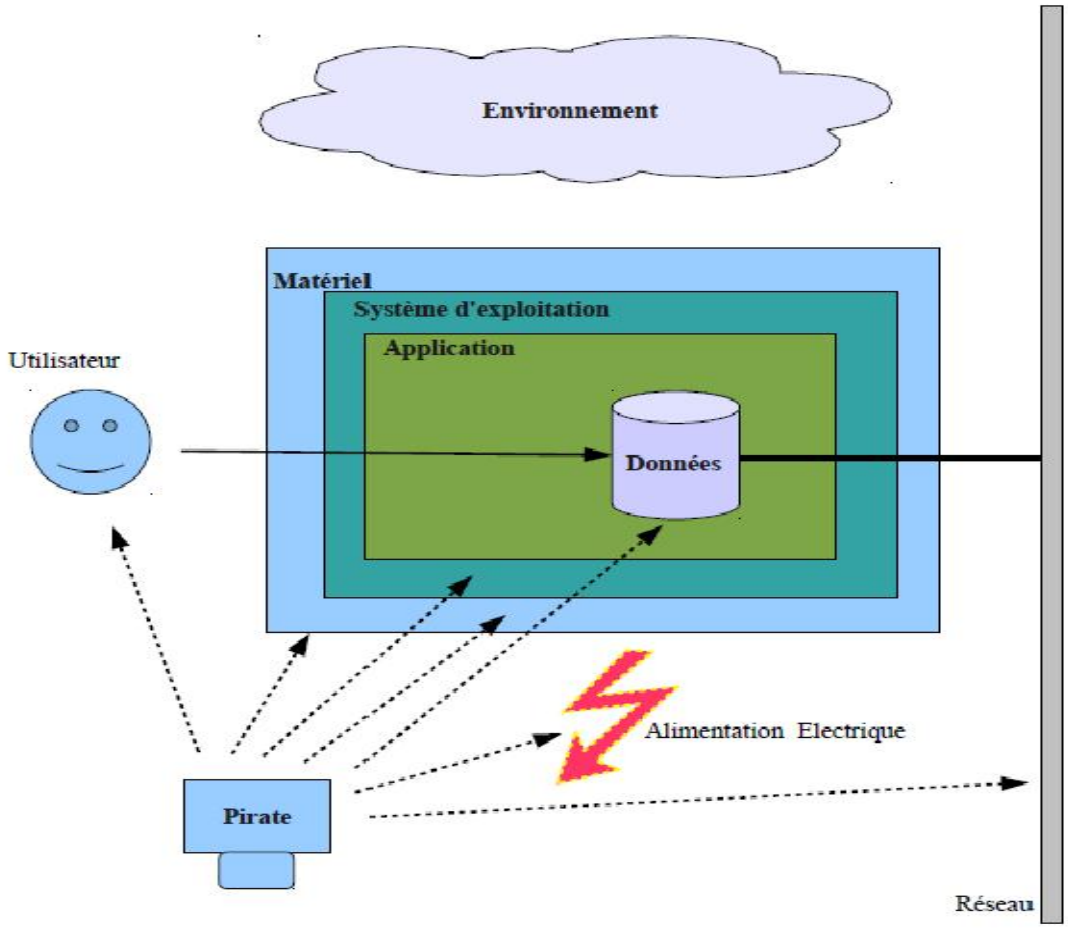

Figure 2.1 : Sécurité et Risque aux Différents Niveaux

2.3. Failles dans la Sécurité des Réseaux

Dans un réseau informatique, une faille dans la sécurité peut être définie comme un «trou » , un disfonctionnement ou une insuffisance dans le dispositif de sécurité existant . Mais d'où peut venir la faille dans la sécurité ? Il existe deux réponses possibles :

- La faille peut venir de l'intérieur du réseau (on peut parler de « faille active »)
- La faille peut être provoquée depuis l'Extérieur du réseau (on peut parler de « faille passive ») .

Failles actives

On peut parler de faille active lorsque celle-ci vient directement de l'intérieur du réseau. « Active » car elle est provoquée par l'action des utilisateurs internes du réseau.

Elle peut résulter de plusieurs facteurs :

➔ Non respect de la politique de sécurité définie dans le réseau, par les utilisateurs internes. Les exemples sont nombreux : installations de programmes non autorisés sur les postes clients ; tentatives de violation de niveau d'accès aux ressources ou services du réseau.

➔ Erreurs de manipulations commises lors du traitement de certains fichiers.par exemple fichiers de configuration des services

Failles Passives

On peut parler de faille Passive lorsque celle-ci est provoquée depuis l'extérieur indépendamment des actions des utilisateurs internes du réseau. « Passive » par ce qu'elle résulte d'une action extérieure au réseau.

Elle peut être le résultat de plusieurs actions telles que :

➔ L'envoi de programmes malveillants, espions ou de virus vers les machines cibles (du Réseauinterne). Ces actions ont pour objectif final de fournir des informations dont a besoin le pirate pour passer à l'attaque. Exemples : Le Cheval de Troyes qui est un virus. Il peut

encapsuler d'autres virus qu'il libère une fois qu'il est introduit dans la machine cible ou le système. Les virus libérés peuvent attaquer des programmes machines, détruire des fichiers ciblés

La détection d'une faille dans la sécurité d'un réseau informatique a pour étape suivante l'exploitation de cette faille. L'exploitation de cette faille constitue en quelque sorte le début de la véritable Intrusion réseau.

2.4. Définition d'une Intrusion Réseau

Une intrusion réseau consiste à se connecter sur un ordinateur ou un réseau distant sans en avoir la permission. Donc c'est la pénétration frauduleuse d'unsystème.L'intrusion réseau est une des conséquences potentielles de l'existence de failles dans la sécurité d'un réseau informatique. Le but (ou encore actions consécutives) d'une intrusion dépend des motivations du pirate (l'intrus) :

- Accès aux documents confidentiels
- Vols ou destructions de données
- Espionnage : introduction dans le réseau de programmes capables de surveiller les

Activités internes du réseau au profit du pirate. Ce qui est très dangereux, car cela peut permettre au pirate d'avoir un control total sur le système

- Dénis de service : il constitue à empêcher aux utilisateurs légitimes du réseau

D'exploiter au maximum les ressources de leurs machines par l'envoi de programmes perturbateurs (bugs) ou des pings continus (Ping de la mort).

2.5. Quelques Techniques d'intrusion

Les Attaques[3][9]

Une intrusion est consécutive à une attaque externe réussie, car avant de pénétrer dans le réseau cible il faut déployer certaines stratégies. On peut répartir les attaques en deux catégories :

- ➔ **Attaques Passives**
- ➔ **Attaques Actives**

Attaques Passives

Une Attaque passive constitue à écouter le trafic du réseau (ou de la machine) cible, donc l'interface de la machine attaquante (du pirate) est en mode écoute. L'objectif est de découvrir et capturer des trames du réseau cible pour y rechercher des informations particulières : clés de cryptages, login et mots de passe, données.

Elle se réalise grâce à des outils tels que : les sniffer, les scanners. Un sniffer est un dispositif, logiciel ou matériel, qui permet de capturer des informations (exemple : trames) qui transitent sur un réseau ou destinées à une machine.

Alors qu'un scanner est un programme qui permet de savoir quels ports sont ouverts sur une machine donnée. Les scanners servent pour les hackers à savoir comment ils vont procéder pour attaquer une machine. Examples de scanners et sniffers :Nmap, whireshark, WiFiScanner, aircrack (wifi).

Attaques Actives

Contrairement à une attaque passive, ici l'attaquant n'est plus en mode écoute .Elles consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau, à interroger le réseau (ou la machine) cible, contourner le dispositif de sécurité existant … par diverses méthodes :

- ➢ **Deny of service (Dos : Déni de Service) :**

Une attaque de type Dos consiste à rendre une machine (ou un réseau) « hors service » en lui envoyant des « programmes perturbateurs ».

- **Les Virus :**

Selon la définition donnée par Fred Cohen, le premier chercheur qui a décrit le phénomène dans une thèse publiée en 1985, un virus est un programme informatique capable d'infecter d'autres programmes en les modifiant afin d'y intégrer une copie de lui-même qui pourra avoir légèrement évolué. A la manière de son frère biologique, il se reproduit rapidement à l'intérieur de l'environnement infecté sans que le porteur (l'utilisateur) en ait conscience. Indépendamment de sa fonction reproductive, le virus contient généralement une charge qui peut causer des dégâts redoutables. Mais il ne peut pas agir de façon autonome car le programme infecté doit être exécuté pour devenir actif.

Une machine peut être attaquée par les virus si elle navigue sur Internet « aveuglément » : téléchargements de fichiers dont la source est inconnue, de logiciels libres, accès à certains sites, ouverture de mails de sources inconnues

Il existe une variété de virus :

- ➔ **Les VIRUS SYSTÈME**
- ➔ **Les VIRUS PROGRAMMES**

2.6. Présentation de Quelques Outils D'intrusion

Après avoir cité quelques attaques nous allons maintenant nous intéresser à certains outils permettant de mener ou préparer des attaques.

Les outils d'intrusion sont des moyens (logiciels) qui permettent de préparer une intrusion .Il en existe une variété. Leurs méthodes d'action varient suivant les concepteurs et les OS sur lesquels ils tournent. Ici Nous nous intéresserons uniquement aux plateformes les plus utilisées : Linux et Windows.

Les outils d'intrusion peuvent être classés en deux familles :

- Les Outils Passifs : les scanners, les sniffer
- Les Outils Actifs

2.6.1. Les Outils Passifs

Ils appartiennent à la famille des scanners. Qu'est-ce qu'un scanner ? : Lorsqu'un serveur offre un service particulier (web, messagerie, mail), il exécute un programme assurant ce service.

Ce programme est en attente de connexions. Les clients devant accéder à ce service doivent connaître l'adresse IP du serveur et le numéro de port associé au service offert.

Sur les systèmes Linux la liste de ces numéros est disponible dans le fichier **/etc/services**.

La plupart des services ont un numéro de port bien défini. Par exemple, un serveur de messagerie utilise le port 25, un serveur web le port 80... Lorsqu'un service est en écoute sur un port, on dit que le numéro de port associé à ce service est ouvert.

L'intérêt du scanner est très simple : il permet de trouver une fois qu'il est lancé, dans un délai très court, tous les ports ouverts sur une machine distante.

Il existe différents types de scanners, certains se contentent juste de donner : la liste des ports ouverts, le type et la version de l'OS tournant sur le serveur, d'autres scanners permettent de tester différentes failles connues sur ces services :

Nmap ,Tcpdump , Netstat , Whireshark.

Nmap

Nmap est un utilitaire qui tourne sous Linux et qui est intégré au système. Nmap donne un aperçu assez complet des différents services s'exécutant sur une machine (même en local) dans un temps assez bref.
Pour connaître les ports ouverts sur une machine, Nmap procède à l'envoi de paquets sur tous les ports de cette machine et analyse les réponses. Il existe différents types de scans, donc différents types d'envois et de réponses. On peut distinguer les scans utilisant les protocoles TCP,UDP. Suivant le protocole indiqué ce sont les informations relatives aux services tournant avec ce protocoles qui sont affichés.

Le scan en mode TCP est le scan par défaut avec Nmap .

La syntaxe basique est :

- **nmap [ip de la machine cible]** →par défaut
- **nmap -sT [ip de la machine cible]** → scan en mode TCP
- **nmap -sU [ip de la machine cible]** →scan en mode UDP

Avec le mode TCP il existe un scan qui est détectable par la machine cible (dans les logs systèmes).

La commande se fait par l'appel de nmap avec l'option **-sS** :

- **nmap -sS [adresse IP de la machine cible]**

Nous présentons ci-dessous, l'utilisation de nmap avec d'autres options :

- **nmapplage_d'adrresse (exemple nmap 192.168.0-255)**→scan sur toutes les machines appartenant à la plage spécifiée.
- **nmap -O [ip de la machine cible]** →pour connaître le système d'exploitation de

La machine cible. Il faut savoir que chaque système d'exploitation construit ses paquets d'une manière bien particulière. Certains champs au niveau de la couche IP ou TCP sont propres à chaque système d'exploitation. Nmap contient une base de données d'un grand nombre de systèmes. Nmap envoie donc des paquets tests à la machine cible et compare les paquets reçus en réponse à ceux de sa base de données et en déduit le type de système. Cette base de données est mise à jour en fonction des différentes version de nmap.

- **nmap -p N° de port [ip de la machine cible] (exemple : nmap -p 80 192.168.0.1)** → scan d'un port précis (ici web).

Wireshark

Wireshark est outil performant permet de capturer et analyser les différents paquets qui circule dans le réseau.

2.6.2. Les Outils Actifs

Ces outils bien que souvent, étant des moyens d'administration réseau, peuvent être utilisées pour attaquer des réseaux à distance. « Actifs » par ce qu'ils ont une vocation « offensive » .

Il suffit juste pour l'attaquant de connaître entre autres l'adresse IP de la cible, le numéro de port sur lequel tourne le service, un compte d'utilisateur ayant accès au service spécifié et son mot de passe. Certains de ces outils sont souvent intégrés au système d'exploitation (Linux, Windows) . Parmi ces outils nous pouvons citer :

Telnet ,ssh , Ping ,Traceroute .

Telnet et ssh

Ces deux protocoles permettent d'attaquer des machines à distance. Ils sont utilisés pour administrer à distance des systèmes informatiques mais peuvent être utilisés à d'autresfins . Une fois que ces deux services (telnet et ssh) sont démarrés chez la machine cible, il suffit seulement au pirate de connaître un login et son mot de passe sur la machine ciblée !Une fois la connexion réussie le pirate peut prendre le contrôle total du système (machine) : il peut se déplacer librement dans l'arborescence, changer le mot de passe de l'administrateur, détruire ou voler des fichiers confidentiels, changer la configuration du système … Sous Windows il existe plusieurs utilitaires permettant de faire du ssh comme **putty** . Pour se connecter par telnet ou ssh il suffit d'être en mode commande et de saisir :

telnet/sshAddresse IP cible (si le numéro de port par défaut n'est pas changé) .

Ping et Traceroute

Un ping permet de vérifier la connectivité d'une machine (pour voir par exemple si elle est en réseau) . Il suffit pour cela de faire : **pingAddresse_IP_cible /Nom d'Hôte .**

Si la cible est en réseau elle va répondre avec des messages **ICMP** sinon on reçoit le message «**Network isunreachable**» .

Cependant il est possible de perturber une machine (Deny Of Service) rien qu 'en lui envoyant une rafale de pings continus de tailles supérieures à la normale (64 bits) : **ping de la mort** , car toute machine a un nombre maximal de requêtes qu'elle peut traiter par unité de temps . Sous Linux l ' option **-w (ping -w durée (en seconde) Addresse_IP_cible)** permet de définir la durée du ping .

Donc ces outils sont très simples mais très efficaces .

2.7. Conclusion

Les techniques de protection contre les attaques Internet permettent de réaliser les bases de la sécurité : confidentialité, intégrité, authentification, disponibilité.

Mais malgré toutes ces techniques utilisées pour empêcher les attaques Internet, un système n'est jamais totalement sûr.

Un système IDS est une nécessité primordiale pour forcer la sécurité de notre réseau

Dans le chapitre suivant, nous présentons l'outil de détection d'intrusion snort.

3.1. Introduction

Snort est un Système de Détection d'Intrusion de réseau Open Source, capable d'analyser en temps réel le trafic sur les réseaux IP.

Snort est capable d'effectuer une analyse du trafic réseau en temps réel et est doté de différentes technologies de détection **d'intrusions** telles que l'analyse protocolaire. Snort peut détecter de nombreux types d'attaques : comme attaque de scans de ports.
Snort est doté d'un langage de règles permettant de décrire le trafic. De plus, son moteur de détection utilise une architecture modulaire de plu-gins.
Snort est principalement dédié aux acteurs de la sécurité réseaux. En effet, sa fonction IDS permet une surveillance des réseaux permettant de détecter et d'alerter en cas de tentative d'intrusion sur le réseau.

3.2. Architecture du Snort [6]

Un noyau de base : (PacketDecoder) 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.
Une série de pré–processeurs: (Détection 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 dessignatures pour les comparer avec la base des signatures.
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.
Une série de « output plugins »: permet de traiter cette intrusion de plusieurs manières envoievers un fichier log, envoie d'un message d'alerte vers un serveur syslog, stocker cette intrusiondans une base de données SQL.

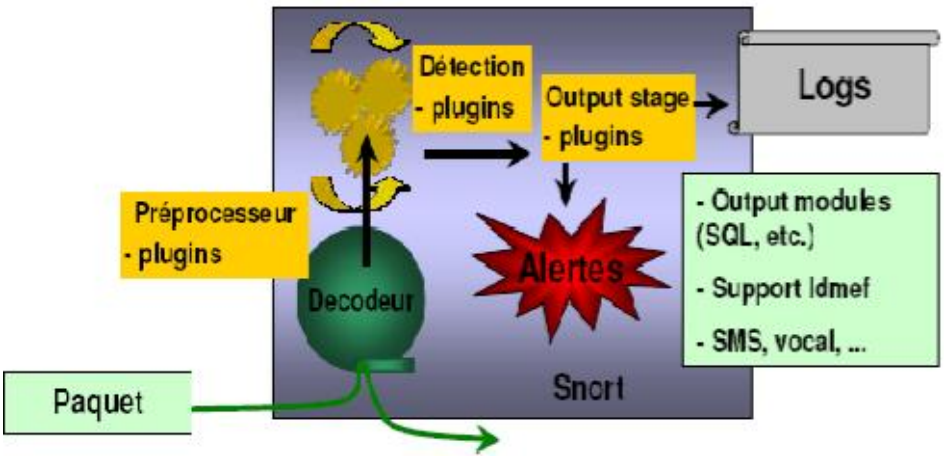

Figure 3.1 : Architecture du snort

3.3. Modes de Fonctionnement de SNORT

Snort peut fonctionner sous trois modes :

- **Le mode sniffer « hors ligne »** qui se contente de lire les paquets qui circulent sur le Réseau et de les afficher de manière continue à l'écran.

Il s'agit d'écouter le réseau, en tapant une ou plusieurs lignes de commandes qui indiqueront à snort le type de résultat à afficher, en voici quelques-unes :

- ✓ La commande verbose affiche les en-têtes TCP/IP : **snort –v .**

L'interface connectée au réseau est automatiquement détectée et scannée. Est-il utile de préciser qu'il faut de l'activité sur le réseau pour avoir des résultats.

- ✓ La commande verbose dump, affiche les IP et les en-têtes TCP/UDP/ICMP : **snort –vde**

- **Le mode « packetlogger »** qui enregistre les paquets sur le disque. Ce mode est en Tout point similaire au précédent, à ceci près que les logs ne s'affichent plus à l'écran, mais s'inscrivent directement dans un fichier de log. Le répertoire naturel de log de snort étant **/var/log/snort/** . La seule modification par rapport à précédemment est le v, remplacé par l, concrètement : **snort -de -l /var/log/snort.** En visitant le répertoire /var/log/snort/ on constatera l'existence de plusieurs répertoires.

- ✓ **Système d'alerte et de Log**

Le système d'alerte et d'enregistrement des logs s'occupe de la génération des logs et des alertes. Dépendant sur ce que le système de détection trouve à l'intérieur d'un paquet, le paquet peut être archivé dans le fichier Log ou une alerte peut être générée. Ces logs sont contenus dans des fichiers.

Dès que le système devient opérationnel, on pourra consulter les alertes générées directement dans les fichiers ou bien utiliser une console de gestion. Il existe des applications qui fournissent une console de gestion et qui permet la visualisation des alertes en mode graphique. Les alertes dans ce cas sont stockées dans une base de données comme mysql comme un titre exemple la base ACIDBASE.

- ➢ **Le mode NIDS (Network Intrusion Détection System)** : le plus complexe et le

Plus configurable, qui permet d'analyser le trafic sur le réseau en suivant des règles définies par l'utilisateur et d'établir des actions à exécuter suivants les cas.

Snort utilise pour cela des règles pour détecter les intrusions. Il existe aujourd'hui environ 1500 règles différentes, chacune s'adaptant à un cas particulier. On peut créer des règles pour observer une activité particulière sur le réseau : pings, scans, faille dans un script, tentative de prise de contrôle à distance. Les alertes peuvent être enregistrées dans un fichier particulier ou directement dans le syslog ou encore dans une base de données. Chaque règle se rajoute dans un fichier de configuration prévu à cet effet, on peut soit utiliser celles qui existent déjà, soit en créer de nouvelles suivant les besoins. Le fichier de configuration de snort est **/etc/snort/snort.conf**, les fichiers **.rules** contenus dans le répertoire **/etc/snort/rules/** sont des fichiers contenant des règles pour un usage bien particulier. Le nom du fichier est, en général explicite, ainsi, **ftp.rules** contient des règles spécifiques au ftp et **dos.rules** s'utilise pour les tentatives de DoS (Denial Of Ser-vice ou Denie de Service en français).

3.4. Les étapes d'installation et configuration Snort[10]

Évidemment nous avons besoin de **Snort**, nous allons configurer Snort pour qu'il enregistre les événements dans une base de données **MySQL.** Nous aurons besoin donc d'installer et configurer le système de gestion de base de données MySQL.

Snort utilise la bibliothèque **Libpcap** pour capturer les paquets transitant sur le réseau, il faut donc s'assurer qu'elle est bien installée. Nous aurons aussi besoin d'une console ou application graphique qui va attaquer la base de données MySQL pour mieux visualiser les alertes et autres informations (statistiques, graphes.).

Nous allons choisir **BASE** qui est une application développée en PHP, mieux c'est une version améliorée de **ACID** nous aurons besoin donc d'installer **PHP**.

Nous aurons besoin d'installer aussi :

- ✓ La librairie de base de données **ADODB**. ADODB est une librairie PHP utilisée pour

Rendre l'accès aux bases de données indépendantes du système de gestion des bases de données utilisé.

3.4.1. Installations des prés-requis

L'installation des prés-requis est souvent délicate. Car les prés-requis dépendent souvent aussi d'autres paquets à installer.

Raison pour laquelle avant d'installer ces prés-requis nous allons faire une mise à jour système pour s'assurer qu'on a au moins des outils de base pour démarrer.

Pour cela on ouvre un terminal et on se connecte en tant que root et on exécute les commandes suivantes :

#apt-get update

#apt-get upgrade

Pour l'installation de certains prés-requis il est plus prudent de faire :

#apt-get install nom_paquet

Ainsi le paquet et ses dépendances seront installés.

Il se trouve que la plus part des prés-requis à installer sont contenus dans d'autres paquets.

l 'installation des dépendances :

#apt-get update

#apt-get upgrade

Pour snort :

#apt-get install libpcap*

#apt-get install libprelude*

Pour mysql :

#apt-get install mysql

#apt-get install phpmyadmin

#apt-get install libmysqlclient15-dev

#apt-get install libpcre3*

#apt-get install libnet1*

#apt-get install libssl-dev

Pour BASE :

#apt-get install adodb*

#apt-get install php5*

#apt-get install apache2*

#apt-get install libapache-mod-php

#apt-get install php-pear

#apt-get install php5-cli

#apt-get install libphp-adodb

#apt-get install php5-gd

Ici nous présentons l'installation des outils essentiels : snort et base.

Installation de snort

Pour l'installation de snort, nous exécutons la commande suivante :

apt-get install snort

Installation de snort-mysql

Après la compilation il nous faut installer le daemon : snort-mysql . Nous allons installer snort-mysql en ligne de commande :

apt-getinstall snort-mysql

Lors de l'installation, une interface graphique s'ouvre :

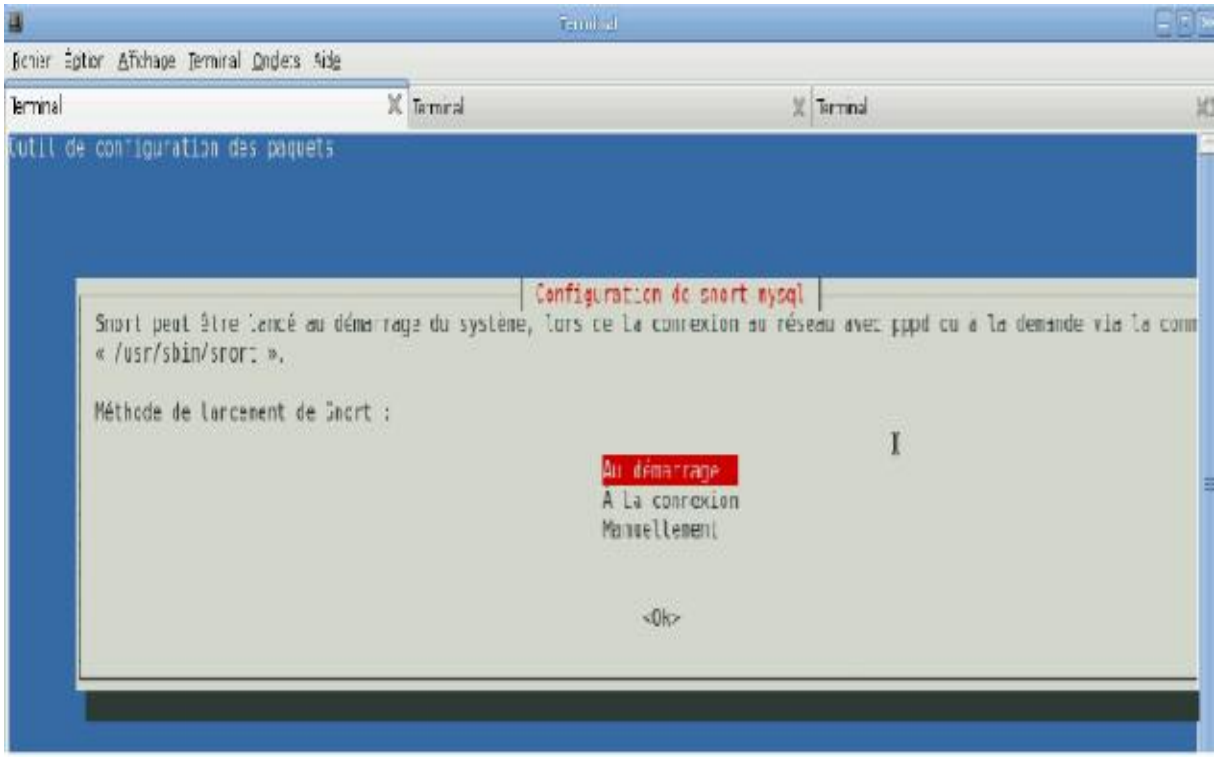

Figure 3.2 : mode de lancement de snort-mysql

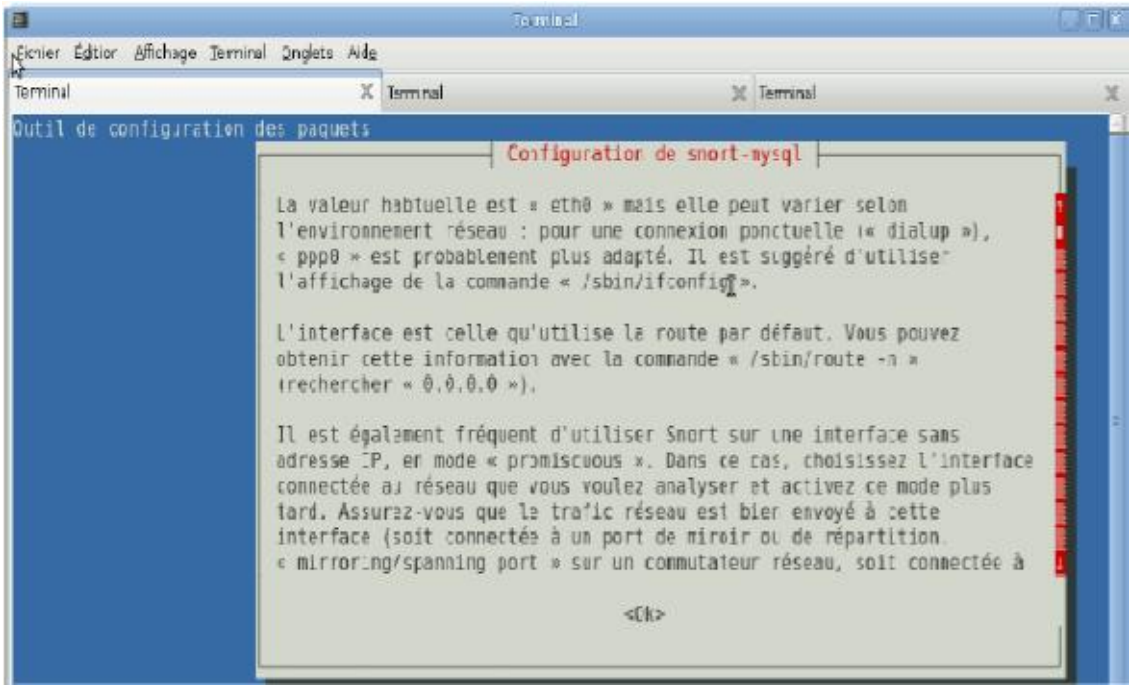

Figure 3.3 : Interface d'écoute de snort-mysql

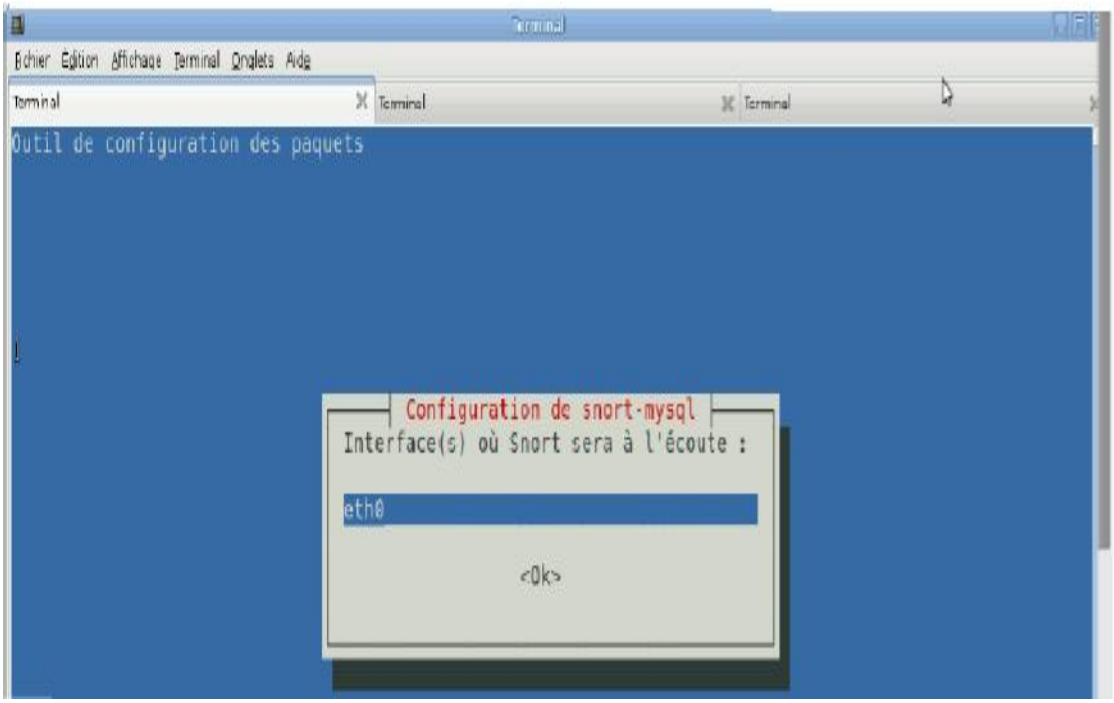

Figure 3.4 : interface eth0 de snort-mysql

Figure 3.5 : désactivation mode promiscuous de snort-mysql

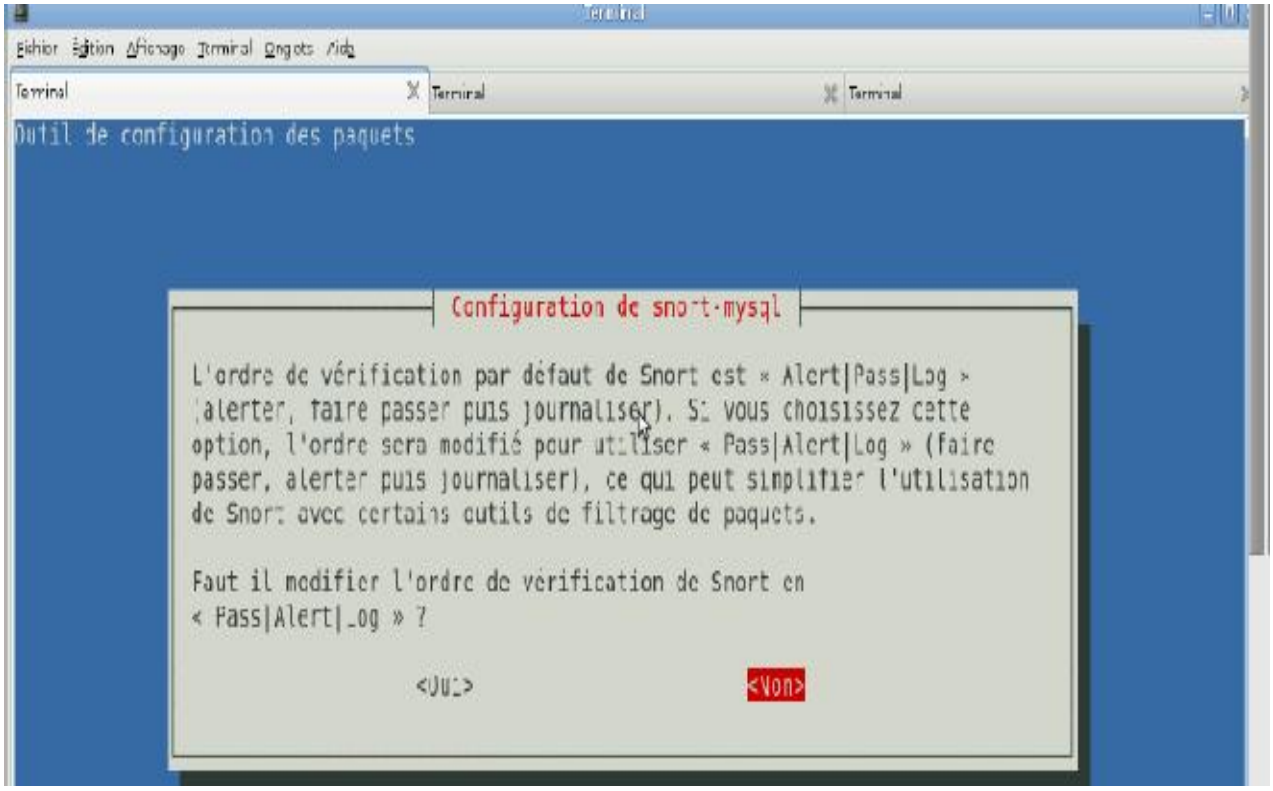

Figure 3.6 : vérification de snort-mysql

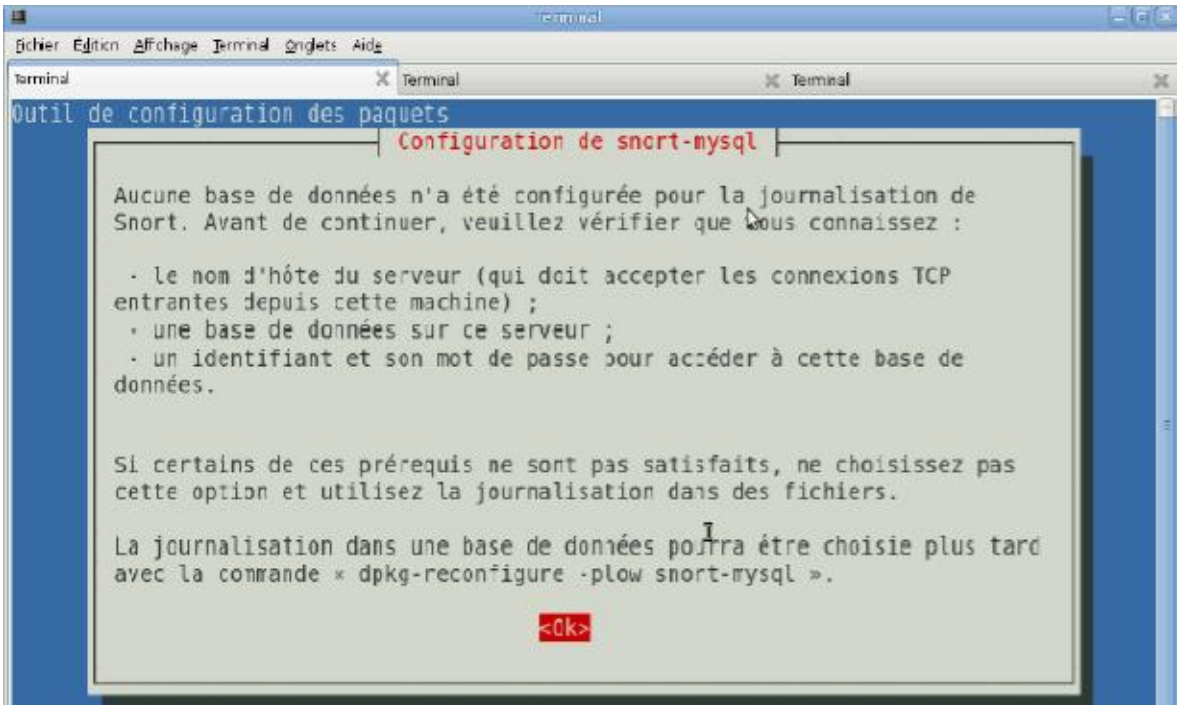

Figure 3.7 : base de snort-mysql

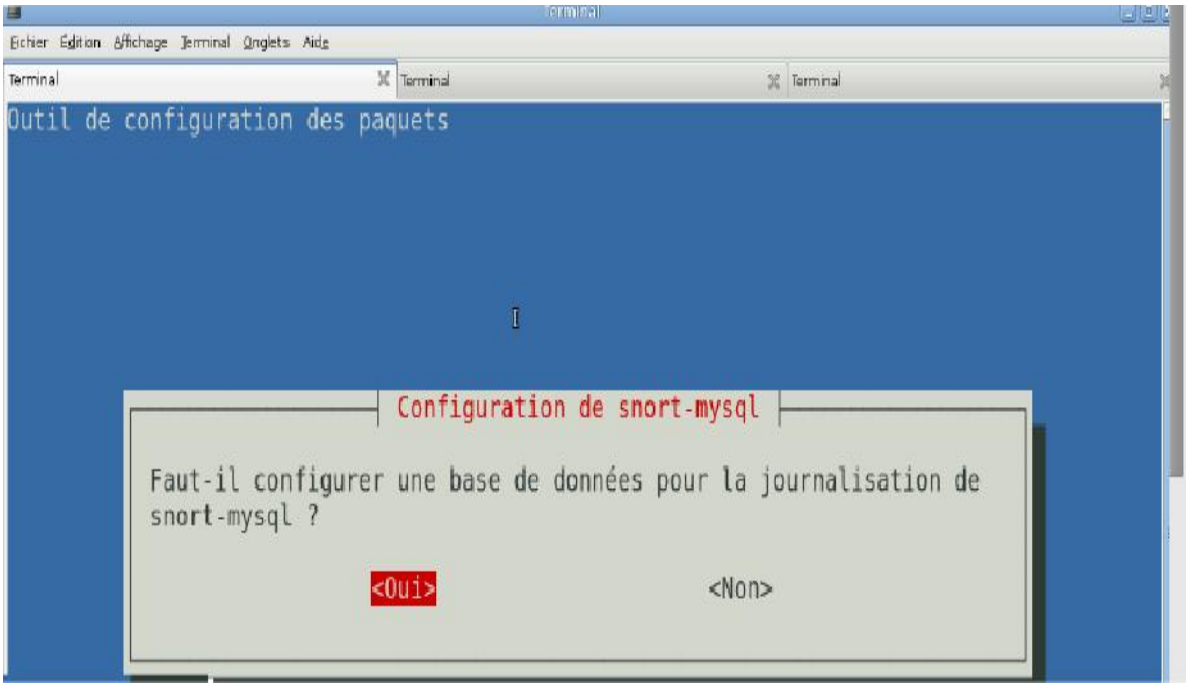

Figure 3.8 : journalisation de snort-mysql

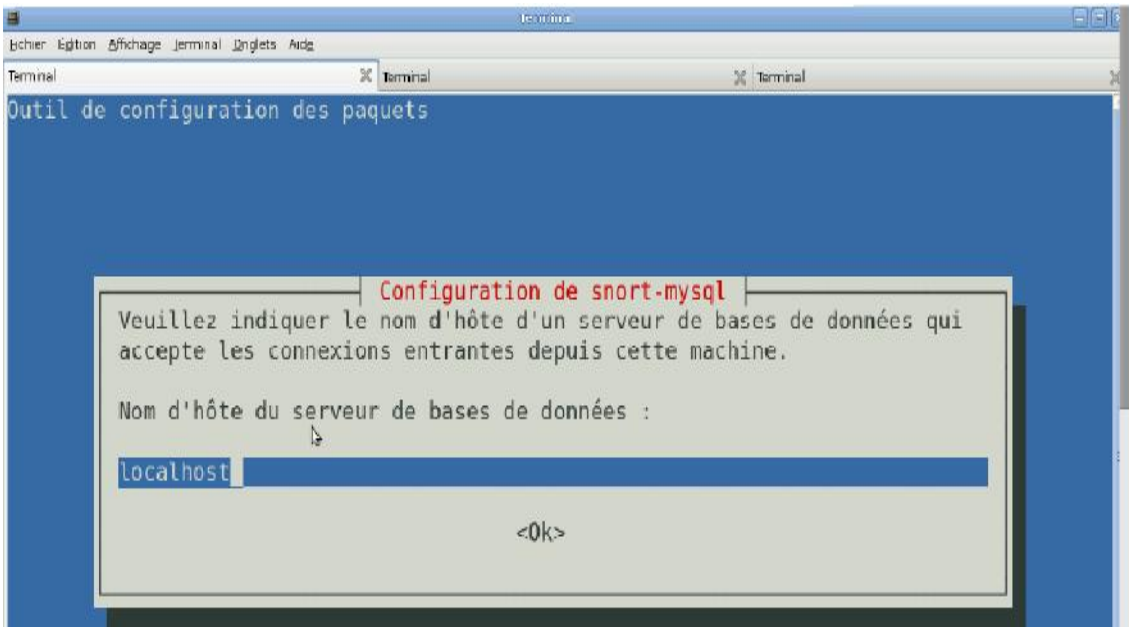

Figure 3.9 : configuration localhost de snort-mysql

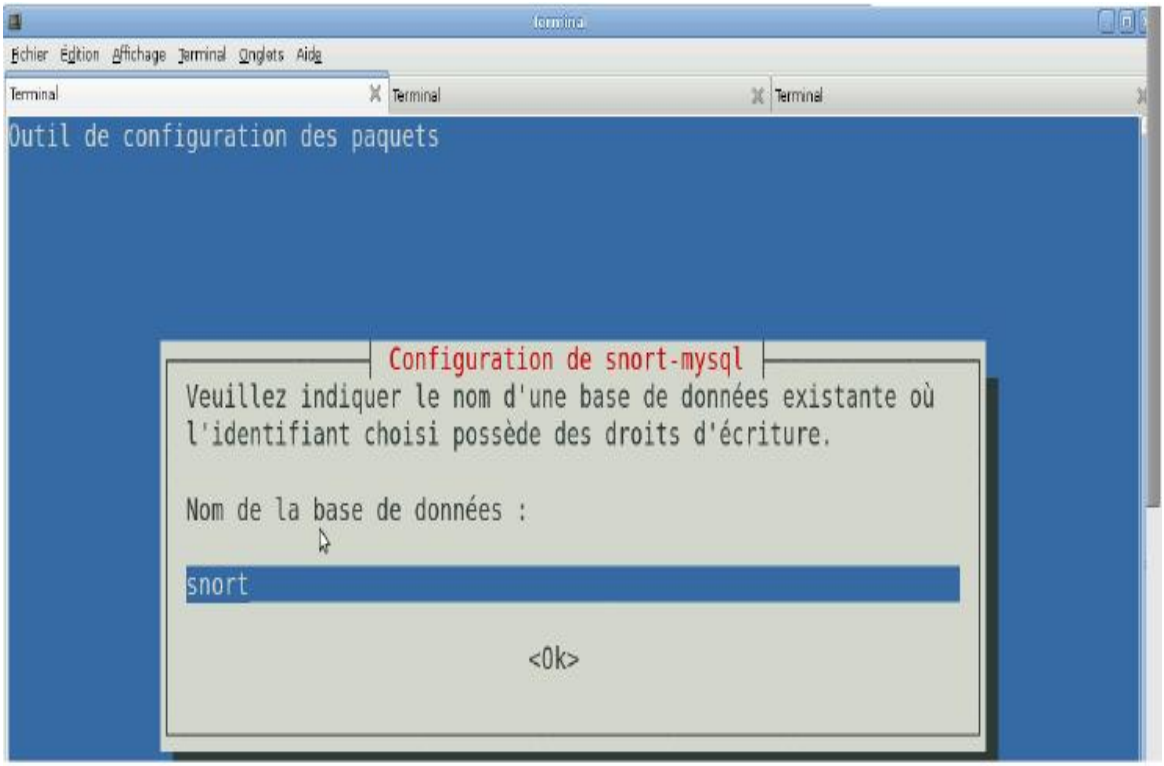

Figure 3.10 : nom de base de snort-mysql

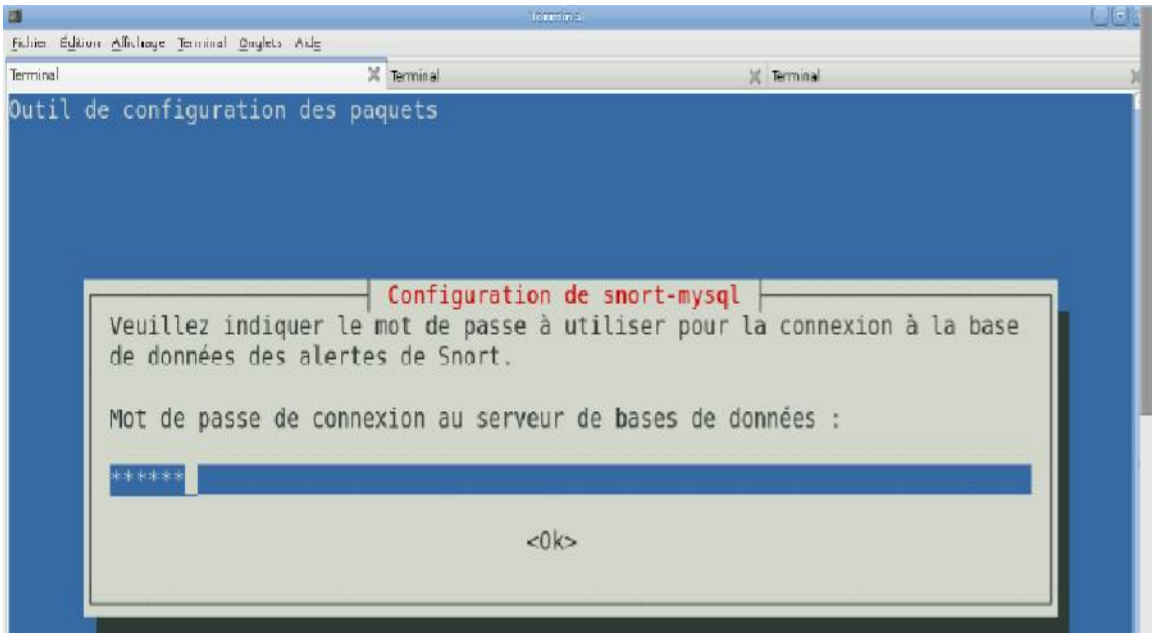

Figure 3.11 : mot passe de snort-mysql

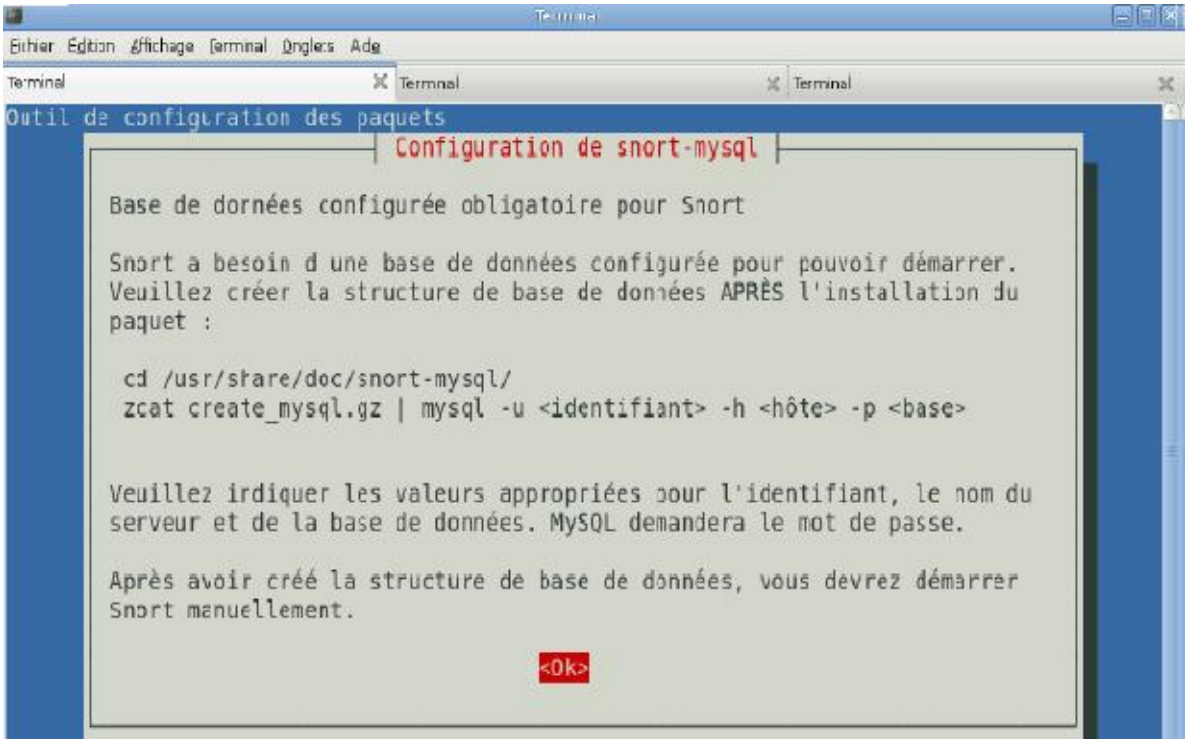

Figure 3.12 : structure de la base de données de snort

3.4.2. Configuration de snort

Nous allons découvrir snort ainsi que ses fichiers de configuration

Snort comporte plusieurs fichiers de configuration. Les plus importants sont : **/etc/snort/snort.conf** (fichier de configuration principal) et les fichiers **.rules** se trouvant dans le répertoire **/etc/snort/rules/** .

Le fichier snort.conf indique à snort dans quel environnement il évolue et quelques paramètres :

- ✓ Le réseau dans lequel il se trouve.
- ✓ Le réseau à surveiller.
- ✓ La liste des serveurs se trouvant dans le réseau où snort est déployé ainsi que leur port d'écoute.

Quant aux fichiers .rules ils contiennent des règles prés-définies. Ces règles déterminent la réaction de snort quand il est lancé. Il existe un fichier appelé **local.rules** qui permet à l'administrateur réseau de définir ses propres règles.

3.4.3. Les règles Snort

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 ainsique 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 certainsChamps du header ou parmi les données.

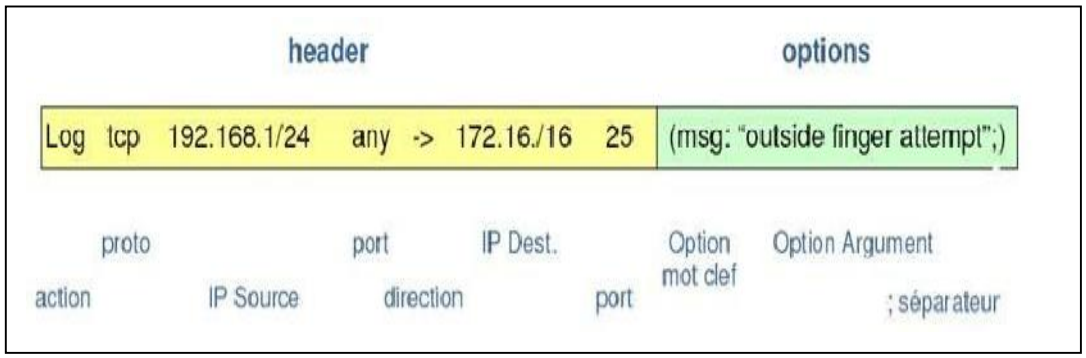

Figure 3.13 : Composition de la règle de snort

Action de la règle: alert, log, pass

Protocole: tcp, udp, icmp
Adresses source et destination: src, dest, any

Port src / dest: any, nb port, plage de ports avec p1:pn

Opérateur de direction: ->unidirectionnel, ou <->bidirectionnel

Syntaxe des options :

- combinaison de règles avec le séparateur « ; »
- séparation des mots clefs et des arguments avec « : »
- mots clefs : msg, logon, minfrag, ttl, id, dsize, content, offset, depth, flags,seq, ack, itype, idecode, nocase, session

Ce schéma ci-dessous récapitule le fonctionnement de snort :

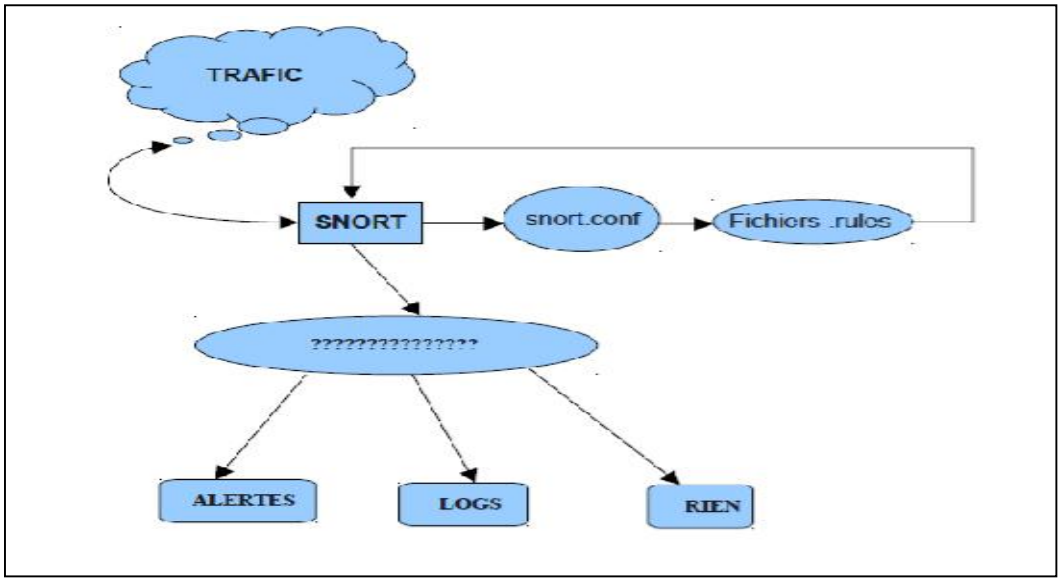

Figure 3.14 : Modélisation du Fonctionnement de Snort

Fichier de configuration Snort

Pour la configuration de snort, nous allons éditer le fichier snort.conf (/etc/snort/snort.conf) . Nous avons uniquement modifié la première partie du fichier qui concerne les variables réseau. Voici les modifications que nous avons apportées dans ce fichier :

- ✓ **var HOME_NET any**/* indique l'adresse de l'interface réseau qui écoute le trafic .

La valeur par défaut est any . On peut personnaliser en mettant l'adresse ip de l'interface ou du réseau à protéger. */

- ✓ **var EXTERNAL_NET any**/* indique le(s) réseau(x) externe(x) à « écouter » * . La valeur par défaut est any , ce qui signifie que le trafic venant de n' importe quel réseau est analysé . Pour exclure le réseau à protéger on met **!HOME_NET** à la place de any. On peut aussi préciser les réseaux en mettant : [adresse réseau1, adresse réseau2 …..]. */

- ✓ **var RULE_PATH /etc/snort/rules** /ici on précise le répertoire où se trouvent les fichiers .rules . */

- ✓ **output database :alert,mysql,user=snort password=pqsserdbname=snort host=localhost .**/* ici on indique à snort qu 'il faut enregistrer les événements dans une base de donnée mysql avec les paramètres ci-dessus */ .

Avant de configurer MySQL, nous allons créer une règle très simple dans local.rules (**/etc/snort/rules/local.rules).** Les règles de snort sont très analogues avec les ACL (Access-List Control) au niveau de la syntaxe et du fonctionnement.

#vim /etc/snort/rules/local.rules

alerttcp any any - > any any (msg : « test »; sid:1000001;)

/* tout trafic TCP venant de n'importe où vers n'importe quel destinataire entraine une alert avec comme titre "test" */ .

Configuration de MySQL

Nous allons configurer MySQL pour stocker les alertes et autres événements générés par snort. Pour cela on se connecte à la base de données MySQL en tant que root :

/#mysql -u root -p

Enter password :

On crée la base de données nommée snort avec la commande :

mysql>**create database snort;**

Query OK, 1 row affected (0.04 sec)

On crée l'utilisateur snort et on lui attribue des privilèges ainsi que ses paramètres de connexion :

mysql>**grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort@localhost;**

mysql>**SET PASSWORD FOR snort@localhost=PASSWORD('[mot_de_passe]');**

mysql>**flush privileges;** /* pour charger les privilèges *./

On peut vérifier tout ceci en faisant :

mysql> use mysql;

mysql>select user,password,host from user where user='snort'.

#mysql -u root -p

Enter password: /* on saisit le mot de passe de root pour mysql */

mysql> use snort;

mysql>show tables;

3.4.4. Installation/Configuration de BASE

BASE est un utilitaire graphique, écrit en php , qui va nous permettre de visualiser à temps réel les alertes et autres informations que snort va «enregistrer » dans la base de donnée MySQL .

Voila l'url de la base ACID : http://localhost/acidbase et on suit les instructions suivantes :

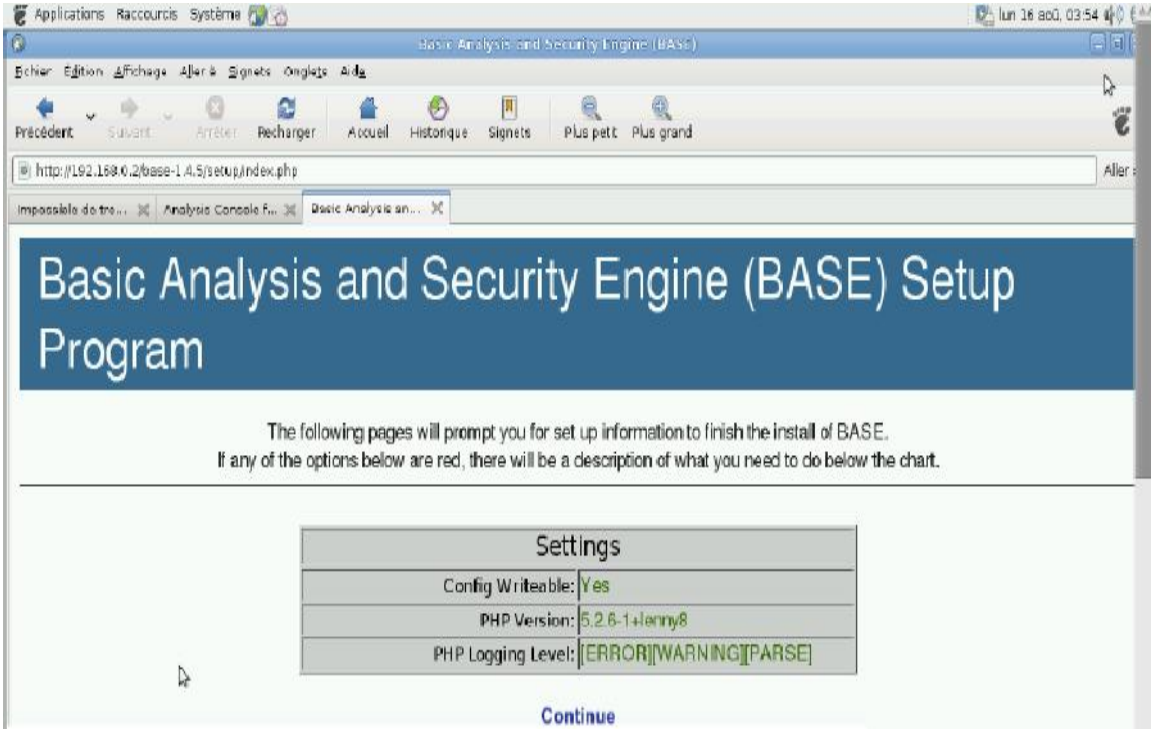

Figure 3.15 : Début d'installation de base

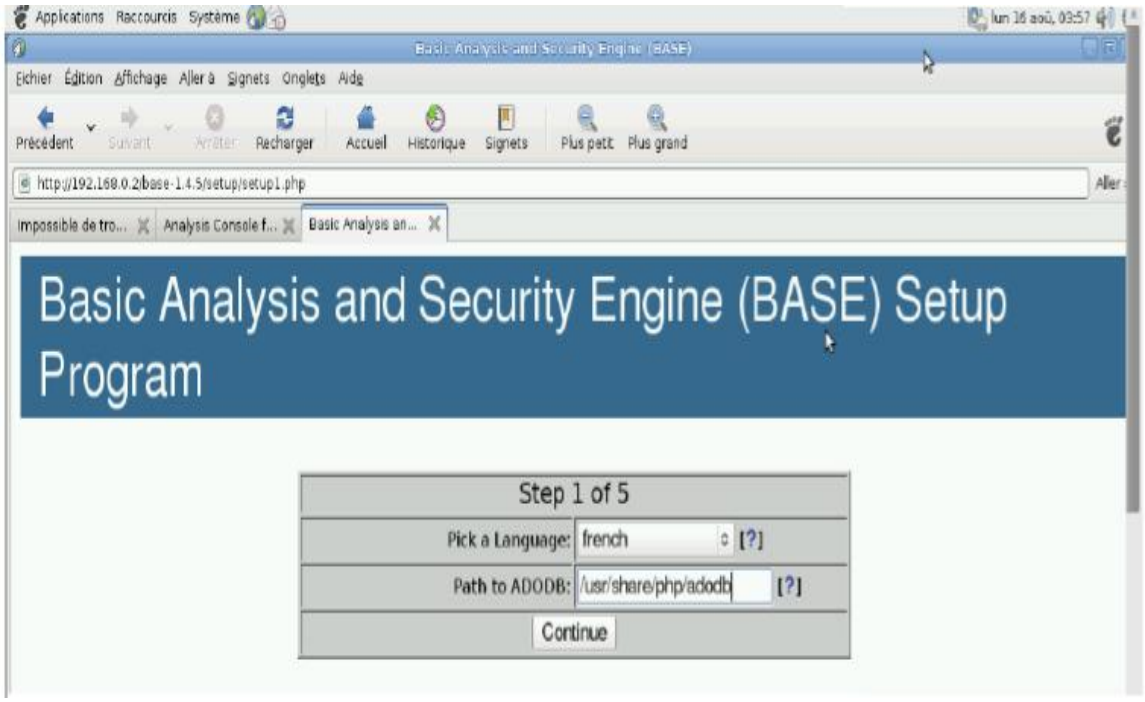

Figure 3.16 : Etape de la configuration de base

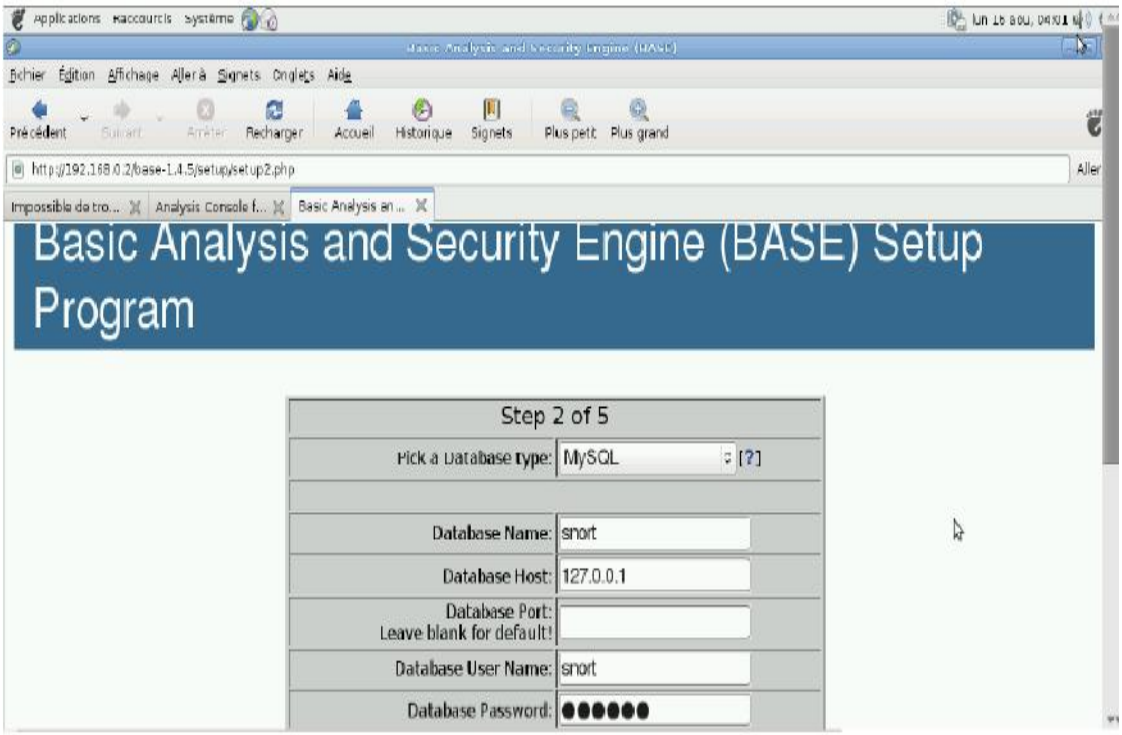

Figure 3.17 : Etape 2/5 de l'installation de base

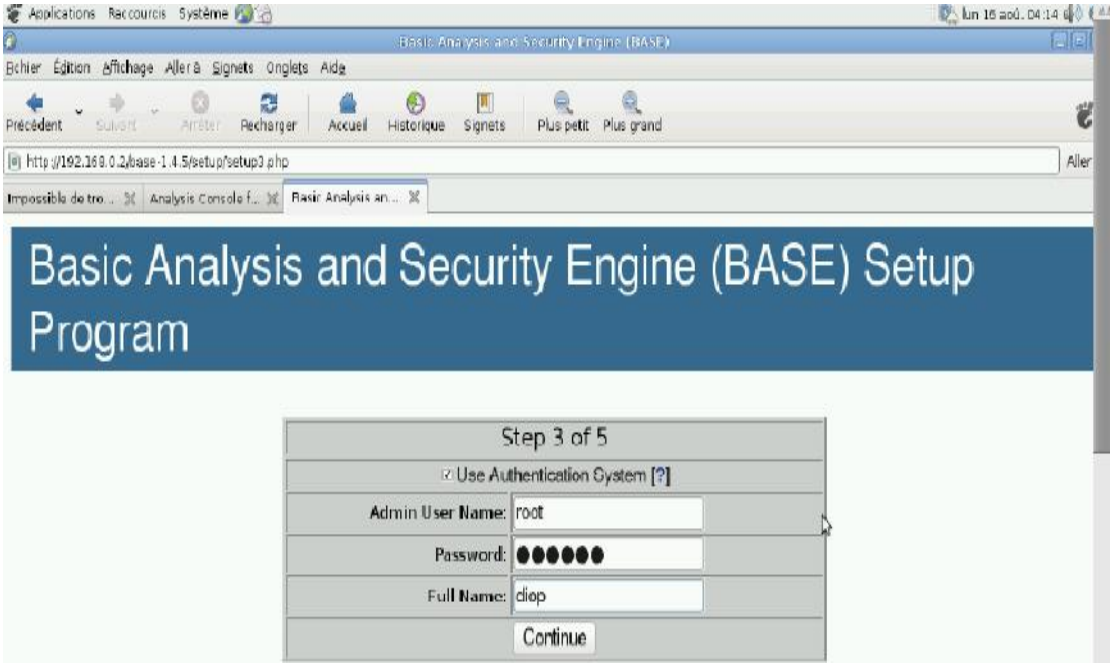

Figure 3.18.: Etape 3/5 de l' installation de base

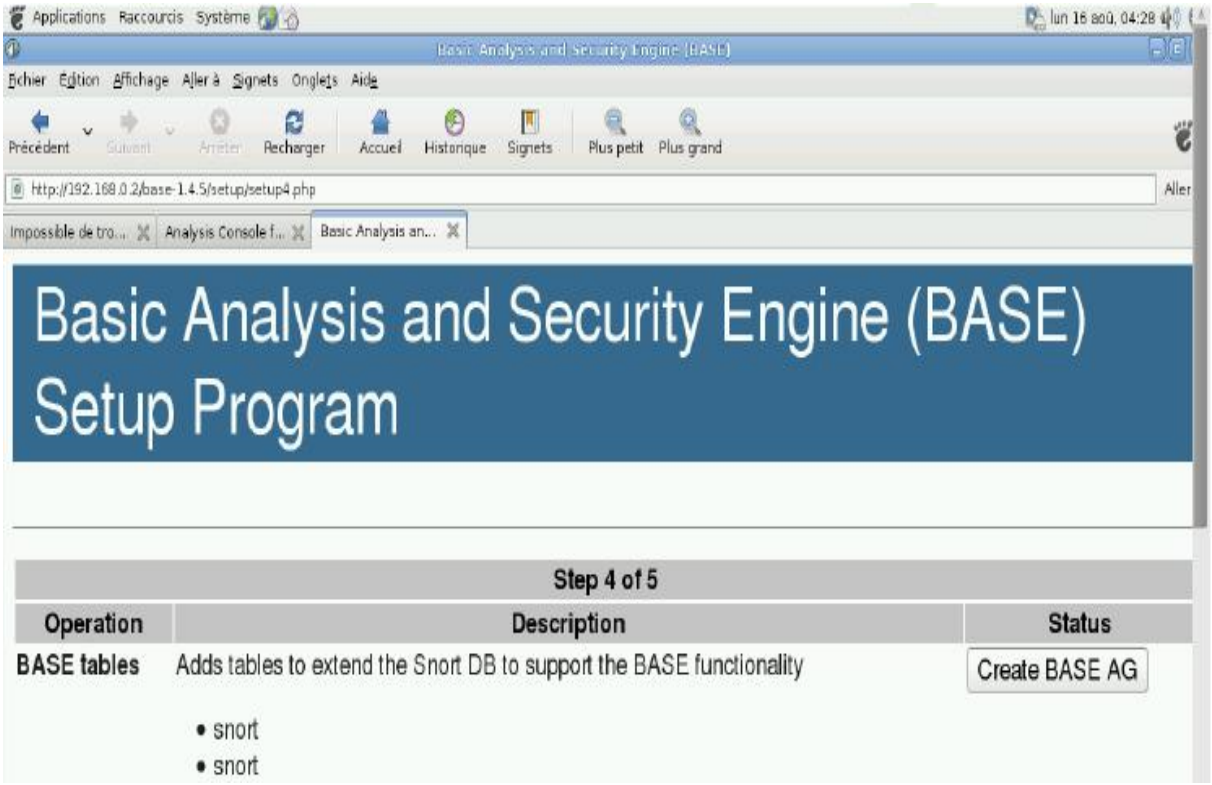

Figure 3.19 : Etape 4/5 de l'installation de base

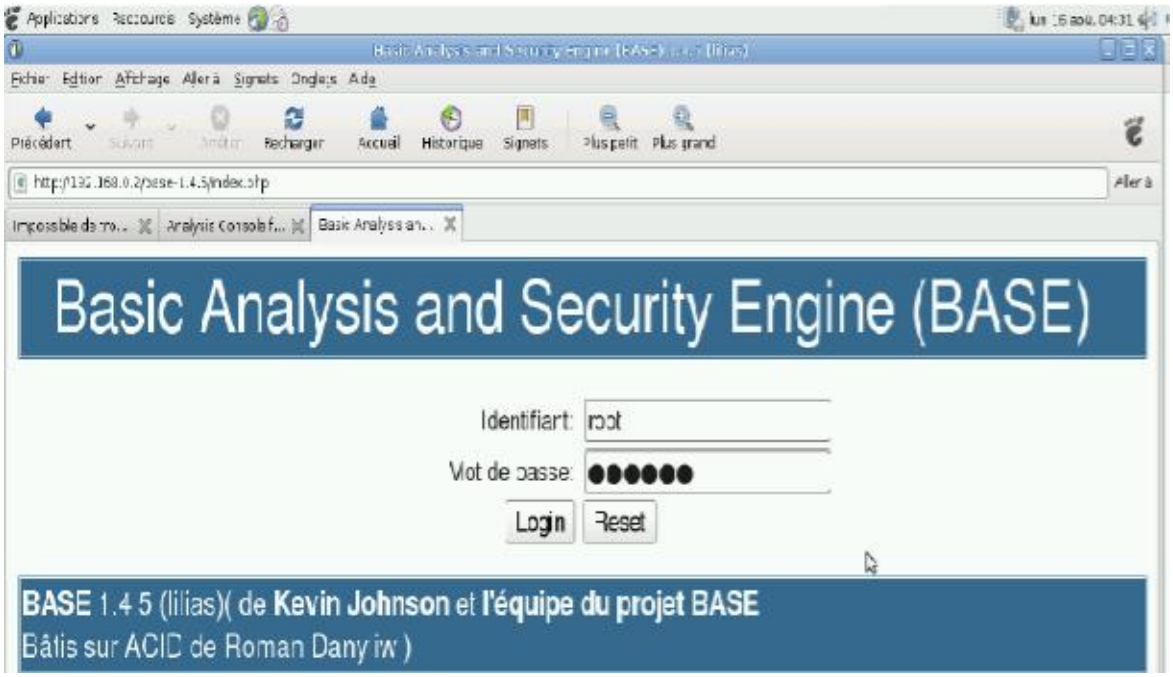

Figure 3.20 : Connexion sur base

À ce niveau l'application base est bien installée. Nous allons nous connecter sur base en tant que root , on peut créer aussi un autre utilisateur différent de root .

Àprés connexion, voici les premières infos qui s'affichent :

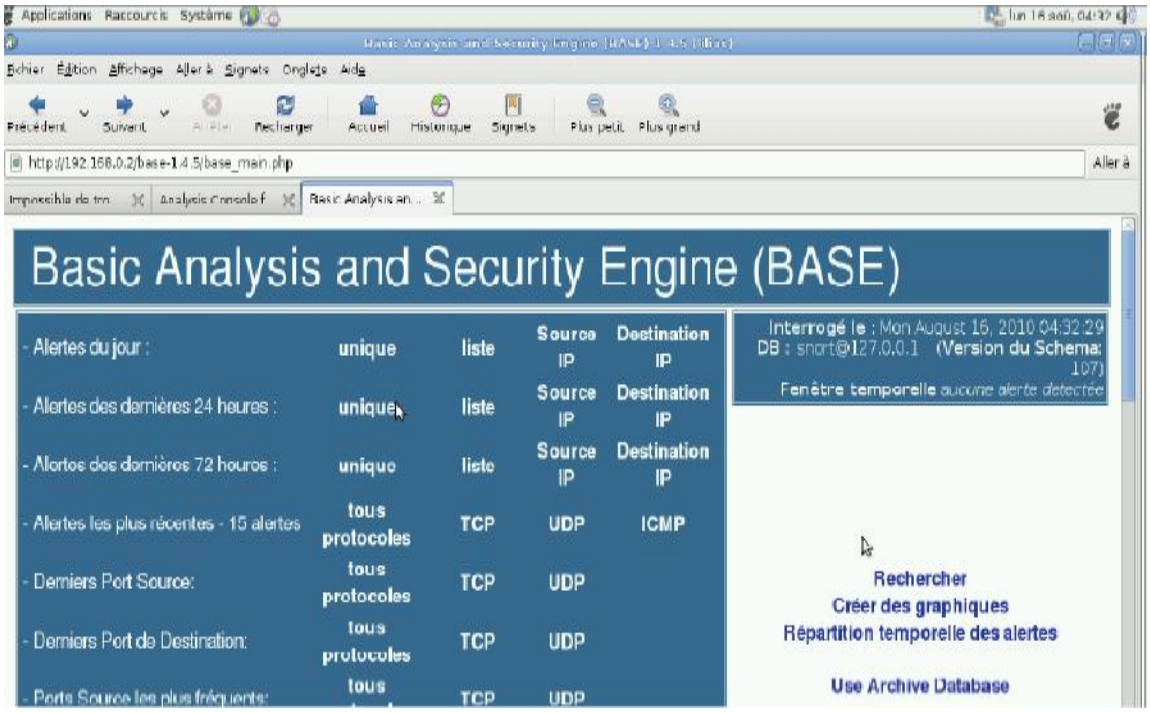

Figure 3.21 : Page d'accueil après connexion sur base

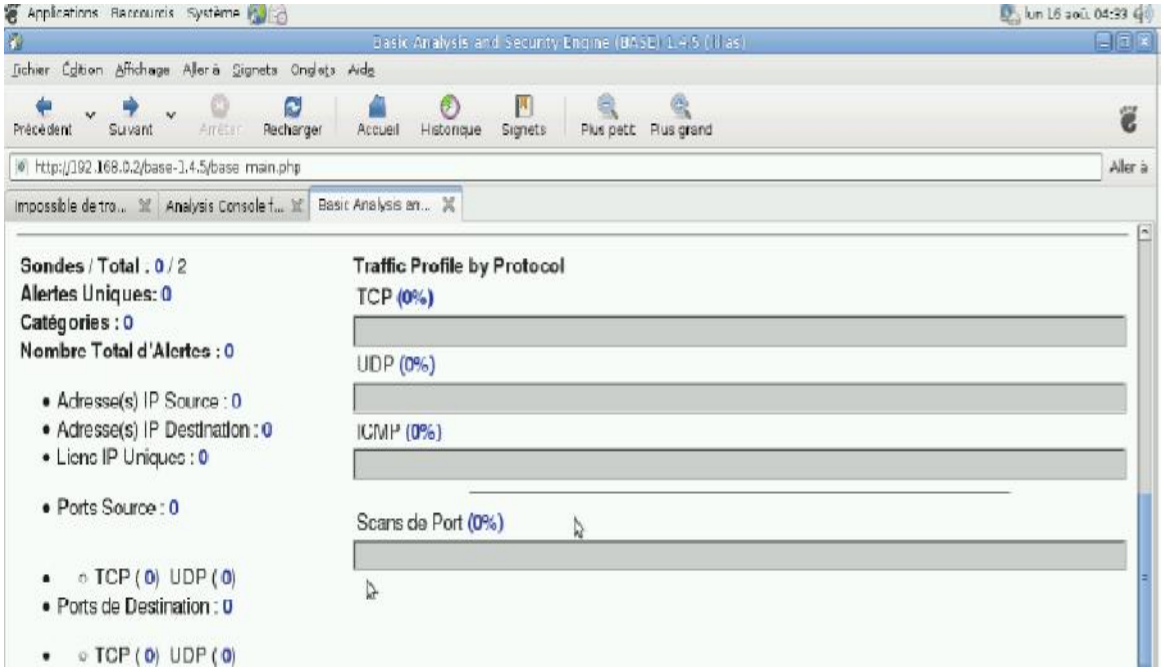

Figure 3.22 : Page d'accueil après connexion sur base (suite)

3.4.5. Utilisation de snort

Dans cette partie nous allons tester snort, puisque tout le nécessaire a été installé. nous ne pourrons faire des tests permettant de simuler à 100% les actions des hackers. Nous allons limiter nos tests aux services ou application de base tels que : l'interdiction de pings , de telnet , du web.vers notre machine censée être la cible (en même temps elle joue le rôle de sensor). Pour les rés ultats que nous allons observer, en réalité tout se passe comme si un inconnu a réussi à traverser le firewall pour faire des pings continus, telnet, ssh ,web …

3.4.6. Modes de fonctionnement

snort peut fonctionner en plusieurs modes :

- **Mode sniffer** : il écoute le trafic et capture les paquets. Quelques informations

Contenues dans les paquets sont affichées. Il est lancé en ligne de commande avec l 'option -v (snort -v) . D'autres options sont possible en faisant man snort .

- **Mode packetlogger** : il est similaire au précédent sauf qu ' ici les résultats sont

envoyés dans des logs (/var/log/snort) . Toujours en ligne de commandes : **snort -de -l /var/log/snort** .

- **Mode NIDS** : en mods nids snort est lancé avec la commande :

snort -u snort -c /etc/snort/snort.conf . Il analyse les paquets et les soumet aux règles se trouvant dans /etc/snort/rules/ . C'est ce mode qui nous intéresse et qui fera l'objet de tests .

Maintenant que snort.conf est bien configuré, nous allons créer nos propres règles dans le **/etc/snort/rules/local.rules**. Nous allons interdire les pings , le web, le ssh,le ftp vers notre sensor qui en même temps joue le rôle de serveur cible .

Voici les règles :

alerticmp any any -> $HOME_NET any (msg:"alert-ICMP";sid:1000003;)

alerttcp any any -> $HOME_NET 80 (msg:"alert-HTTP";sid:1000004;)

alerttcp any any -> $HOME_NET 21 (msg:"alert-FTP";sid:1000006;)

alerttcp any any -> $HOME_NET 22 (msg:"alert-SSH";sid:1000007;)

alerttcp any any -> $HOME_NET 23 (msg:"alert-TELNET";sid:30000;)

.Pour les tests voici les opérations que nous avons effectuées :

Puis nous ouvrons un navigateur et on saisit l 'url pour se conecter au serveur web et ftp : http://192.168.0.2 ; ftp://192.168.0.2 :

Figure 3.23. Test 2 (http vers la cible 192.168.0.2)

Figure 3. 24 : Test 3 (ftp vers la cible 192.168.0.2)

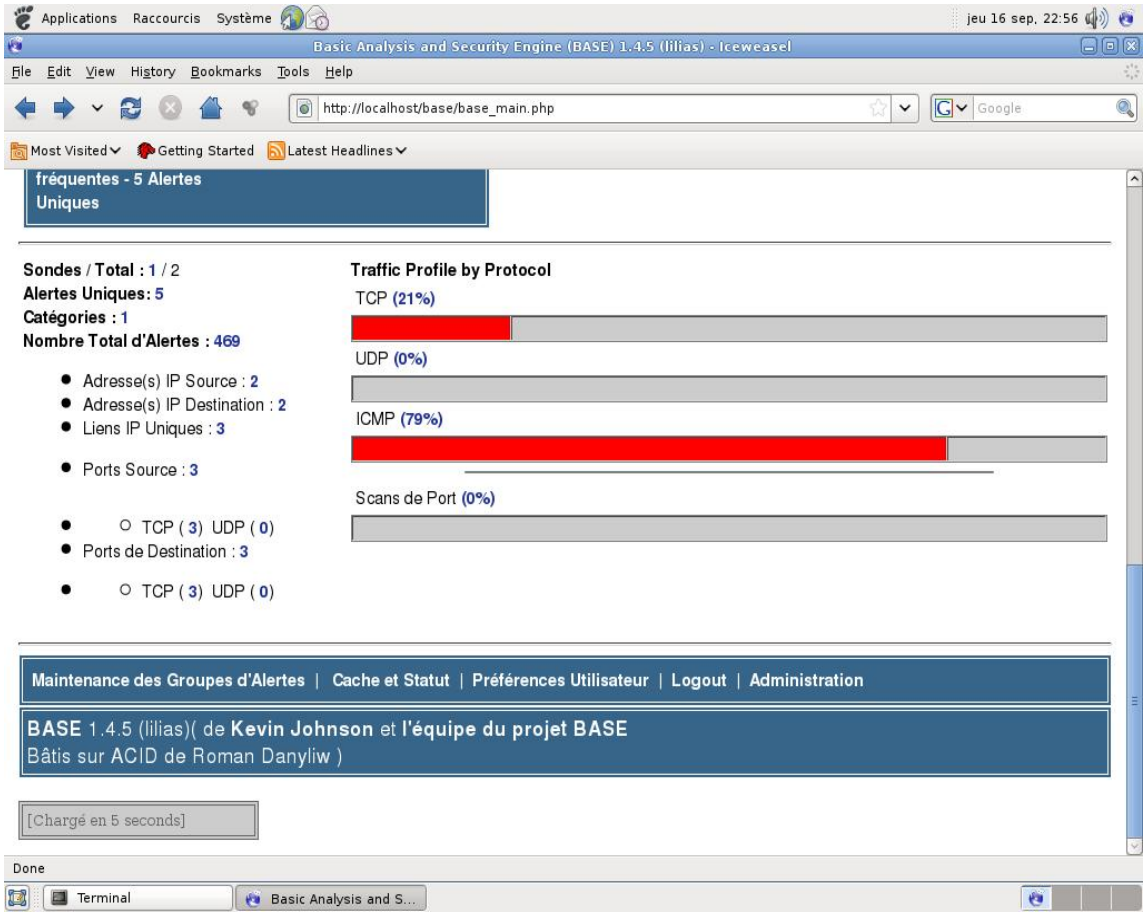

Figure 3.25 : Résultats enregistrés par base après les tests

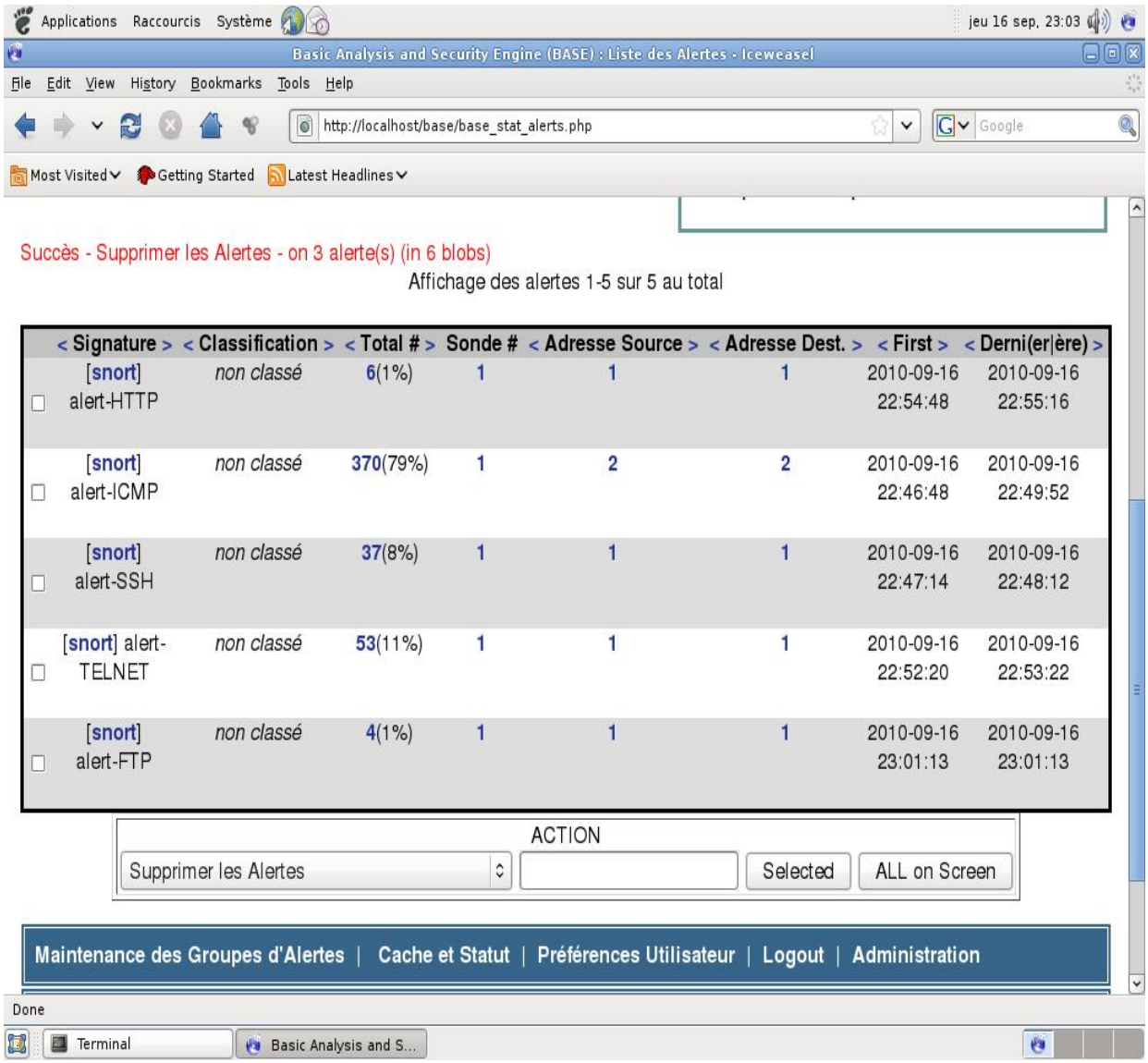

Figure 3.26 : Alertes générées par snort

Donc snort a détecté des actions qui étaient définies comme une intrusion. Et on remarquera aussi que trop d'alertes ont été générées par snort. Dans les conditions réelles il serait difficile de gérer toutes ces alertes.

Pour pallier à cela il faut choisir des règles « non sensibles » dans les fichiers .rules.

3.5. Conclusion

Nous avons présentons dans ce chapitre un outil important pour la détection d'intrusion concernant l'outil Snort. Nous avons donnés touts les étapes d'installation et configuration de cet outil.

Les systèmes de détection d'intrusion, en particuliers snort, peuvent être assimilés à de simples alarmes qui se déclenchent une fois qu' 'ils découvrent une intrusion.

Conclusion générale

La sécurité des réseaux informatiques demeure encore et toujours un sujet très sensible voire complexe, pour les acteurs du monde informatique, car les variables qui tournent autour de ce sujet sont souvent difficiles à maitriser. Même si l'évolution de la technologie a permis

D'améliorer les mécanismes de sécurité dans les réseaux informatiques, il est toujours difficile voire impossible de garantir une sécurité à 100%. Ce qui fait que la sécurité dans le monde des réseaux sera toujours un bras de fer entre les innovateurs (chercheurs, experts …) et les hackers. À toute innovation, les pirates tenteront de la contourner.

Ce projet de PFE aura été pour nous,une grande percée et un long apprentissage dans l'aspect sécurité réseau. Il nous aura permis aussi de nous familiariser avec l'environnementLinux, qui est loin d 'être "hospitalier" !

Les systèmes de détection d'intrusion, en particuliers snort, peuvent être assimilés à de simples alarmes qui se déclenchent une fois qu'ils découvrent une intrusion.

Ce qui constitue une limite notoire pour l'IDSsnort. Mais cela ouvre des pistes de réflexion pour mettre en place des outils ou mécanismes "post-détection",c'est à dire des outils à vocation « actives » (permettant par exemple de bloquer la connexion de la machine source » en cas d'intrusion.

[1] : http://web.mit.edu/rhel-doc/OldFiles/4/RH-DOCS/rhel-sg-fr-4/ch-detection.html

[2] : http://securinet.free.fr/intrusions.html

[3] : http://securinet.free.fr/attaques.html

[4] : http://www.tux-planet.fr/utilisation-de-nmap-et-outil-de-detection-des-scans-de-ports/

[5] : http://www.linux-france.org/prj/edu/archinet/systeme/ch05s03.html

[6] : http://www.sestream.com/docCom/Snort.pdf

[7] : http://www.ossir.org/resist/supports/cr/200205/ids-logs.pdf

[8] : http://www.aubeuf-hacquin-yoann.fr/realisations/presentationstage3.pdf

[9] : http://www.labo-linux.org/articles-fr/surveillance-reseau-avec-snort/installation-et-configuration-de-snort

[10] : shttp://www.system-linux.eu/index.php?post/2009/02/23/Compilation-Installation-et-Configuration-de-Snort

[11] : http://blog.nicolargo.com/2007/06/comment-surveiller-son-reseau-avec-le-cochon-snort.html

[12] : http://www.mi.parisdescartes.fr/~osalem/enseignement/DU/SNORT/TP_SNORT_DU.pdf

[13] : http://file.trustonme.net/documentation/187.pdf

[14] : http://irt.enseeiht.fr/anas/cours/tp-ids2.pdf

[15] : http://www.fichier-pdf.com/telecharger-ebook-snort-gratuit-convertir-pdf-2.htm

[16] : http://ws.edu.isoc.org/workshops/2005/ccTLD-Dakar/jour3/securite/security-full-fr.pdf

[17] : http://irt.enseeiht.fr/anas/cours/tp-ids3.pdf

[18] : http://www.hynesim.org/files/652/Session2k4.honeypot.presentation.pdf

[19] : http://cg.scs.carleton.ca/~mathieu/MCouture_SARLalonde2004.pdf

[20] : http://www.learningtree.fr/courses/588.pdf

[21] : http://linuxgateway.free.fr/Solution%20Linux%202006%20Correlation%20et%20IPS.pdf

[22] : http://www.rennes.supelec.fr/polux/SLIDES/Thomas_28_09_2007.pdf

[23] : http://www.rince.fr/IMG/pdf/DMR200305120550036.pdf

[24] : http://www.fullsecurity.ch/fileadmin/documents/sims/sims-webJwinteregg.pdf

[25] : http://www.ssi.gouv.fr/IMG/cspn/anssi-cspn-cible_2009-06fr.pdf

[26] : http://www.tc2l.ca/fileadmin/tc2l/pdf/INFOGLOBE_Solutions_de_securite_GNU_Linux.pdf

[27] : http://placid.insa-rouen.fr/-Livrable16.pdf

[28] : http://www.alcove.fr/IMG/pdf/white_paper_firewall.pdf

[29] : http://www.student.montefiore.ulg.ac.be/~fryns/rapport_groupe1.pdf

[30] : http://intertrack.naist.jp/FRA_Tutorial-InterTrack-Config.pdf

[31] : http://www.enib.fr/~harrouet/Data/Courses/coursSniffSpoof.pdf

[32] : http://lehmann.free.fr/RapportMain/node11.html

[33] : http://www.linuxfrance.org/prj/inetdoc/securite/tutoriel/tutoriel.securite.attaquesprotocoles.ip-spoofing.html

[34] : http://www.securiteinfo.com/conseils/introsecu.shtml

[35] : http://www.commentcamarche.net/contents/attaques/usurpation-ip-spoofing.php3

[36] : http://www.commentcamarche.net/contents/attaques/attaques.php3

[37] : http://www.ysosecure.com/types-attaques/attaques-securite.asp

[38] : http://www.le-webmaster.com/informatique/attaques.php

[39] : http://wiki.backtrack-fr.net/index.php/Wireshark

[40] : http://ditwww.epfl.ch/SIC/diode/ssh.html

[41] : http://www-igm.univ-mlv.fr/~dr/XPOSE2004/IDS/IDSSnort.html