Table des matières

LISTE DES TABLEAUX ET DES FIGURES

TABLE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION

Avec le développement des réseaux et de l'Internet, le partage de l'information a gagné de plus en plus de popularité et d'importance. Dans ce contexte assurer la sécurité des systèmes informatiques constitue un enjeu crucial. En effet, au cours de ces dernières années la sécurité des systèmes informatiques a pris de plus en plus d'ampleur. Chaque jour nous découvrons de nouvelles attaques, toujours plus sophistiquées, et de virus potentiellement dangereux pour les systèmes et réseaux informatiques. Ceci montre qu'il est d'actualité de trouver sans cesse de nouvelles parades à ces attaques pour assurer la sécurité des ressources informatiques.

Les Systèmes de Détection des Intrusions (IDS: Intrusion Detection System) constituent une des moyens les plus utilisés pour contrer les attaques. La fonction principale des IDS consiste à repérer les tentatives de nuire à un système informatique de par le matériel et/ou le logiciel. La détection d'intrusion consiste à scruter le trafic réseau, collecter tous les événements sur un hôte, ou sur un ensemble d'hôtes, les analyser et générer des alarmes en cas d'identification de tentatives d'action malveillante. Il existe toutefois plusieurs techniques permettant de les prendre à défaut.

L'objectif du présent mémoire est de d'améliorer les capacités des IDS à résister aux attaques les plus fréquentes. A cet effet nous nous proposons d'étudier les IDS afin d'identifier leurs vulnérabilités, qui sont mises à profit dans les différentes attaques. Plus particulièrement nous nous intéressons aux attaques exploitant les vulnérabilités du protocole ARP. Ensuite nous proposons un mécanisme permettant à un IDS de contrer ce type d'attaque. Pour atteindre les objectifs fixés nous avons divisé le présent travail en trois parties.

Dans la première partie nous étudions les principes généraux de la sécurité dans les réseaux et systèmes informatiques. Ensuite nous examinons les menaces sur la sécurité, les attaques les plus courantes et les solutions proposées pour les contrer.

La deuxième partie de ce mémoire est consacrée à l'étude des systèmes de détection d'intrusions. Nous présentons l'architecture de base d'un IDS, les différents types d'IDS existants, leurs avantages et inconvénients, ainsi que leur positionnement selon la structure du réseau.

Dans la dernière partie nous étudions le protocole ARP et ses insuffisances. Ensuite nous proposons un mécanisme permettant à l'IDS Snort de résister aux attaques passant par ARP. Enfin une plate-forme de test est proposée, afin de valider la solution proposée.

Première partie

La sécurité dans les réseaux et systèmes informatiques

Première partie : La sécurité dans les réseaux et système informatiques

I.1 Les Réseaux et Systèmes Informatiques

L'apparition des micro-ordinateurs dans les années 80, et du modèle OSI (Open Systems Interconnection) en 1977 ont favorisé l'utilisation de l'informatique dans plusieurs domaines d'activité (militaires, commerciaux, médicaux, éducations …). Ce qui va susciter le besoin de partage de données par plusieurs utilisateurs et entraînant ainsi la naissance de nouveaux réseaux en utilisant les lignes téléphoniques donnant ainsi l'Internet.

Cette merveilleuse avancée technologique a permis de faciliter la communication et surtout le partage de l'information par plusieurs utilisateurs quelle que soit leur position géographique. Mais cette avancée a malheureusement engendré des risques majeurs dans la sécurité des données.

I.2 Gestion des réseaux informatiques

Un réseau informatique est une entité très complexe. Il propose des services très diversifiés, multiformes et renferme un grand nombre d'équipements variés (ordinateurs, routeurs, commutateurs …etc). Dans ces conditions, la gestion de son fonctionnement s'avère une lourde tâche, d'autant plus qu'un réseau est souvent étendu et rassemble des technologies hétérogènes.

Un réseau peut être encore défini comme un ensemble de nœuds reliés entre eux par des liens.

- Un nœud peut être : une station de travail, un micro-ordinateur, un routeur, un modem, une imprimante …etc ;
- Un lien peut être : un câble métallique (paire torsadée ou câble coaxial), une fibre optique, des faisceaux hertziens …

Pour assurer son bon fonctionnement ou ses interconnexions, le réseau a besoin d'équipements tels que les passerelles, les routeurs, les commutateurs et les concentrateurs (hubs). Il existe deux architectures de base pour les réseaux : le modèle OSI et le modèle TCP/IP.

I.2.1 Le modèle OSI

OSI (Open Systems Interconnection) ou Interconnexion de Systèmes Ouverts, est un modèle à 7 couches [1.5]. Il constitue aujourd'hui le socle de référence pour tous les systèmes de traitement de l'information. Ces différentes couches sont représentées dans la figure1.1.

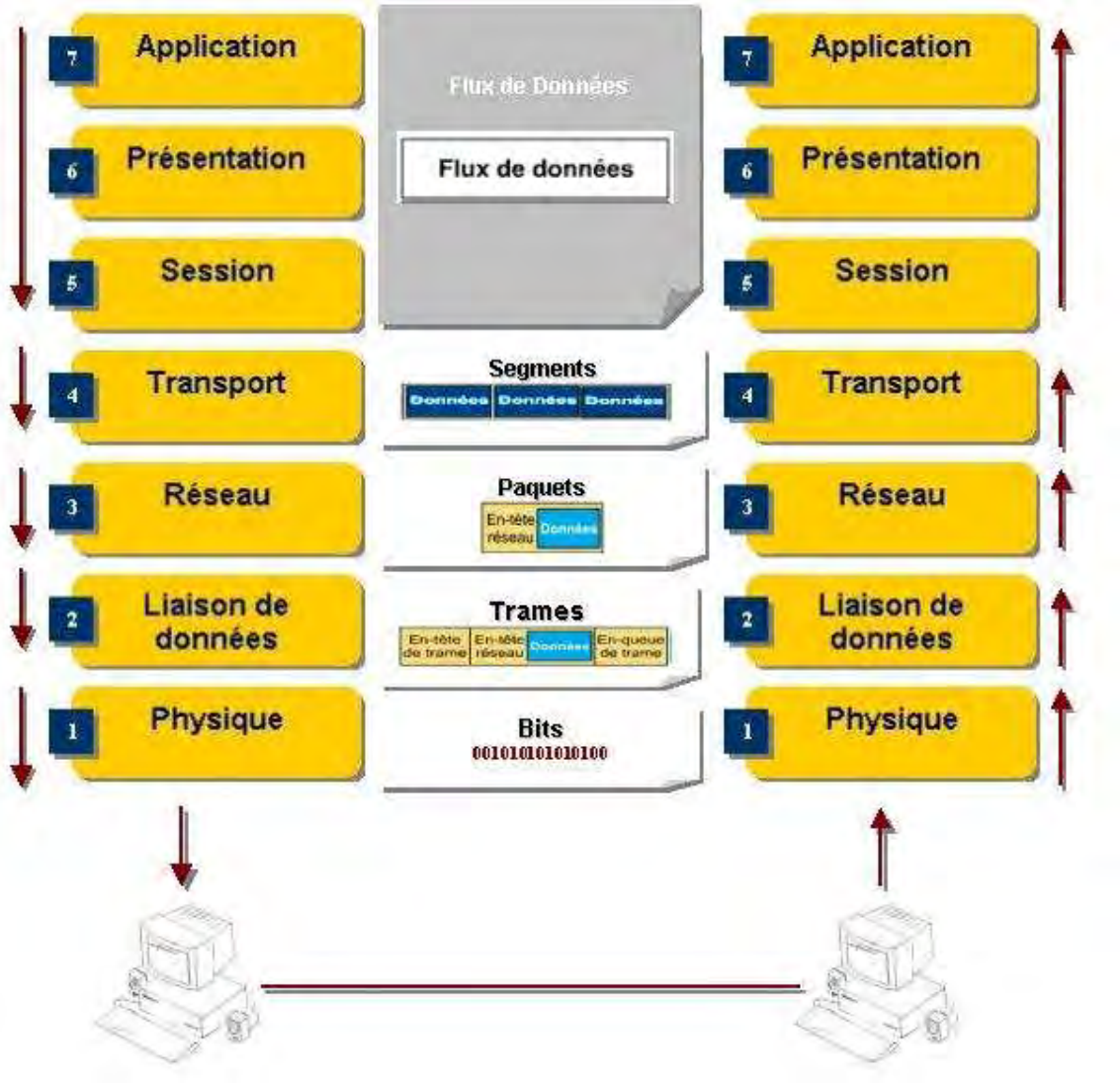


Figure 1. 1: Architecture du modèle OSI

Un paquet envoyé parcourt, du sommet à la base de cette pile de couches. Chaque couche ajoute ses propres en−têtes au paquet, ce que nous appelons le processus d'encapsulation. Lorsque le paquet rejoint sa destination il parcourt en sens inverse la pile de couche et les en−têtes sont supprimés du paquet, un à un. Chaque en−tête donne à l'hôte de destination toute l'information nécessaire, jusqu'à ce que le paquet joigne l'application ou le programme auquel il était destiné. Parmi ces couches on distingue : la couche physique, la couche liaison de données, la couche réseau, la couche transport, la couche session, la couche présentation et la couche présentation.

- **La couche physique :** Cette couche s'occupe de la transmission des données sous un format (binaire, analogique) selon la nature du support de transmission;

- **La couche liaison de données :** Elle joue le rôle de « liant ». Elle fractionne les données d'entrée de l'émetteur en trames, transmet ces trames en séquence, gère les trames d'acquittement renvoyées par le récepteur et gère la fiabilité du transfert des bits d'un noeud à l'autre du réseau ;
- **La couche réseau :** La couche réseau, détermine les routes de transport et s'occupe du traitement et du transfert de messages. L'unité de donnée est le paquet ;
- **La couche transport :** La couche transport regroupe les règles de fonctionnement de bout en bout, assurant ainsi la transparence du réseau vis-à-vis des couches supérieures. Elle traite notamment, l'établissement des connexions et la fiabilité du transport ;
- **La couche session :** Cette couche, s'occupe de l'établissement, de la gestion et coordination des communications. Elle réunit les procédures de dialogue entre les applications : établissement et interruption de la communication, cohérence et synchronisation des opérations ;
- **La couche présentation :** Cette couche s'occupe de la mise en forme des données, éventuellement de l'en cryptage et de la compression des données;
- **La couche application :** Cette couche fournis les services aux applicationx qui ont besoin de communiquer par le biais du réseau. C'est donc le point de contact entre l'utilisateur et le réseau.

I.2.2 Le modèle TCP/IP

L'architecture TCP/IP est presque similaire à celle du modèle OSI, mais le modèle TCP/IP repose sur 4 couches. On distingue la couche accès réseau, la couche Internet, la couche transport, et la coucheapplication. Ces dernières peuvent être représentées sur la figure 1.2

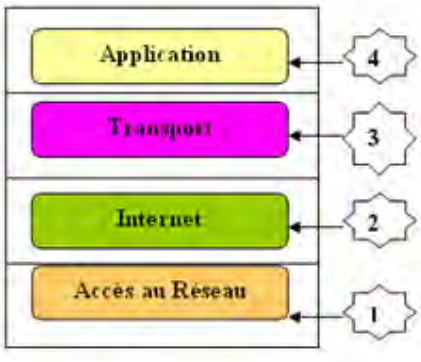


Figure 1. 2 : Architecture du protocole TCP/IP

Dans le modèle TCP/IP, 4 couches sont suffisantes pour décrire l'architecture de ce protocole. C'est donc une architecture compactée du modèle OSI, c'est-à-dire certaines couches du modèle TCP/IP correspondent à plusieurs couches du modèle OSI.

- **Couche accès réseau:** Elle spécifie la forme sous laquelle les données doivent être acheminées quel que soit le type de réseau utilisé ;
- **Couche Internet :** Elle est chargée de fournir les paquets de données (datagramme) ;
- **Couche transport :** Elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaître l'état de la transmission ;
- **Couche Application :** Elle joue le rôle d'intermédiaire entre les processus utilisateurs et le réseau.

Ces deux architectures peuvent être résumées dans la figure 1.3

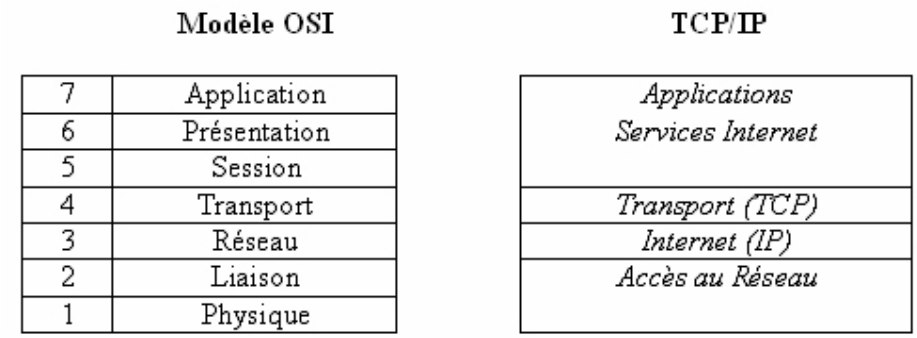


Figure 1. 3 : Correspondance entre OSI et TCP/IP

Ces deux architectures de base (OSI, et TCP/IP) des réseaux informatiques sont multicouches et chacune des couches offre des fonctionnalités équivalentes. Les plus grands avantages de ces couches sont qu'elles offrent une architecture globale et très extensible, d'autres nouvelles fonctionnalités peuvent se greffer facilement aux couches sans avoir à ré-implémenter l'ensemble du code TCP/IP [1.6].

Comme ces protocoles sont relativement faciles à mettre en place, ils permettent d'interconnecter plusieurs types de réseaux, ce qui a favorisé leur développement rapide donnant ainsi le grand réseau actuel de l'Internet, qui désigne un ensemble très vaste et couvrant toute la planète, un réseau dans lequel des machines reliées entre elles selon ces protocoles, et chacune pouvant communiquer avec toutes les autres.

I.3 La sécurité dans les réseaux et systèmes informatiques

Avec le développement de l'Internet et l'ouverture des entreprises, qui ne cesse de croître, à leurs partenaires et/ou à leurs fournisseurs, la baisse du prix de l'ordinateur et la baisse de la tarification pour la connexion sur Internet, le monde est devenu de nos jours un village planétaire où l'accès à l'information est devenu plus facile et plus aisé. Cette merveilleuse avancée technologique a malheureusement engendré beaucoup de trous de sécurité dans les réseaux et systèmes informatiques.

Il est donc nécessaire pour le responsable informatique (administrateur système et réseau) de bien connaître les enjeux liés à la sécurité, et relatifs aux ressources de l'entreprise. Il lui appartient d'identifier les risques et d'analyser les dangers potentiels courus par le système d'information et leurs probabilités d'occurrence. Il se doit d'évaluer leurs répercussions sur les fonctions critiques de l'entreprise et de soumettre les analyses effectuées à sa direction, qui décide, avec son conseil, de leur classement en niveaux majeurs ou mineurs. Enfin, une importante part de son travail doit être consacrée à la sensibilisation des utilisateurs et à leur encadrement. C'est pourquoi l'approche de la sécurité informatique doit être globale, c'est à dire qu'il faut tenir en compte de tous les composants de notre système d'informatique : les éléments matériels, les personnes physiques et la logique informatique.

I.3.1 Les éléments physiques ou matériels

Les éléments physiques sont constitués essentiellement de tous les matériels : ordinateurs, disques de sauvegardes, commutateurs, routeurs ...etc. Le choix des éléments et de leur nombre est le plus souvent conditionné par un budget annuel voté par l'entreprise. Il faut donc qu'il y ait convergence entre les couples 'sécurité/technologie' et 'qualité/prix'.
Pour donc bien sécuriser les éléments physiques et matériels il faut limiter leurs accès physiques en sécurisant les locaux où ils sont logés.

I.3.2 Les personnes physiques

Les personnes physiques sont constituées de l'ensemble des utilisateurs du système disposant ou non de responsabilité et/ou de droits. Ces utilisateurs doivent suivre impérativement les ordres et

les recommandations du chef de la sécurité qui gère le système d'informations. Ils doivent coopérer pour remonter les problèmes et questions qui se présenteront dans l'exploitation des services offerts par le réseau. Pour cela une formation et une sensibilisation sur les questions de sécurité (les précautions à prendre, et la conduite à tenir) doit être faite :

- le choix d'un bon mot de passe (minimum 8 caractères) comportant des majuscules, des minuscules, et des chiffres ou même des caractères alpha numériques: « M0t2!Pass » par exemple est un bon mot de passe très difficile à casser par force brute;
- la confidentialité d'une session, c'est-à-dire ne jamais communiquer ses paramètres de connexions (login et son mot passe) à ses collègues de travail ;
- Toujours fermer sa session, éteindre son poste avant de partir.

I.3.3 Les éléments logiques

C'est la partie la plus complexe à mettre en œuvre. Il faut ici bien schématiser l'emplacement des données, la circulation des informations, les droits aux données, les filtrages, et la sécurité à mettre en place et surtout être paranoïaque en se surprotégeant. Une chose très importante doit être prise en compte, c'est de se tenir informer en permanence des nouvelles failles de sécurité sur les systèmes, de tester et de lire les fichiers logs de son système informatique régulièrement pour ne pas dire chaque jour. Nous étudierons par la suite un certain nombre de moyens mis en oeuvre pour mieux sécuriser les systèmes informatiques.

I.4 Les menaces de sécurité dans les réseaux informatiques

Comme disait le professeur K. SHANKAR[1] du département Informatique et de la Science de l'Information de l'Université Delaware :

"Alors que la société devient de plus en plus dépendante de l'informatique, le crime informatique ne devient plus seulement désastreux dans ses possibilités mais aussi, de plus en plus attrayant"

De nos jours les attaques sur les réseaux deviennent de plus en plus nombreuses et multiformes. Surtout avec l'ouverture et l'interconnexion des systèmes informatiques, des attaques

[1] http://www.tele.ucl.ac.be/EDU/ELEC2920/1997/securite/html_3.htm

exploitant les failles de ces systèmes, et contournant leurs mécanismes de sécurité sont toujours possibles [1.9]. Dans la littérature on parle de deux grandes catégories d'attaques :

- **Les attaques passives :** consistent à écouter le trafic dans un réseau sans modifier les données. Elles sont très difficiles à détecter et constituent une violation de la politique de sécurité. En effet elles sont souvent utilisées à d'autres fins.
- **Les attaques actives :** visent une modification, une insertion ou une destruction des données. Elles peuvent aussi d'empêcher leur bon fonctionnement des équipements réseau. Elles sont donc relativement faciles à détecter.

I.4.1 Le déni de service (Denial of Service: DoS)

C'est une attaque dont l'objectif est d'empêcher le bon fonctionnement d'un équipement, comme par exemple surcharger la bande passante avec l'attaque **UDP Flooding** ou exploiter les vulnérabilités du protocole TCP/IP avec l'attaque **SYN flood**.

L'attaque UDP Flooding consiste à exploiter l'avantage de priorité du protocole UDP sur le protocole TCP en envoyant un grand nombre de paquets UDP. Ce qui va occuper toute la bande passante et ainsi rendre indisponible toutes les connexions UDP.

L'attaque SYN flood, lors d'une connexion TCP normale, le client commence par envoyer une requête SYN au serveur, ensuite le serveur répond au client avec un SYN / ACK, à la réception de cet acquittement, le client envoie un ACK final au serveur. Dans une attaque SYN flood, l'attaquant envoie de multiples requêtes SYN à la victime avec de fausses adresses source comme adresses de retour. La fausse adresse mène à un réseau fictif. Le serveur de la victime répond alors avec un SYN / ACK à un réseau qui n'existe pas, et comme il n'y a aucun récepteur pour recevoir le SYN / ACK, le serveur de la victime attend l'acquittement ACK du client. A cause de cette attente sans résultat, qui consommera les ressources disponibles (mémoire par exemple), le serveur de la victime peut éventuellement ne plus répondre. La localisation de l'attaque est complexe car les adresses contenues dans les paquets SYN envoyés sont falsifiées, on ne peut donc pas déterminer la véritable source. Ainsi si les requêtes se multiplient un DoS est facilement créé.

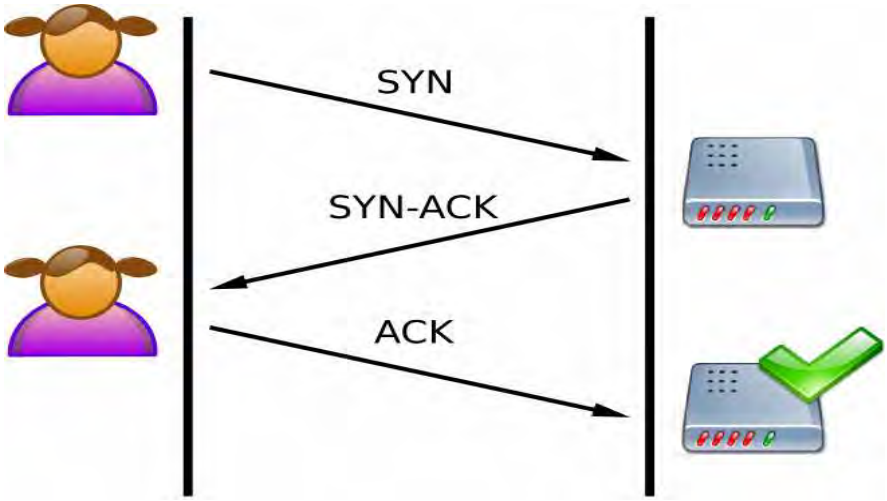


Figure 1. 4 : Connexion TCP normale

I.4.2 Autres types de menaces

Il existe plusieurs autres types de menaces parmi lesquelles on distingue :

- **L'usurpation** ou mascarade : ici l'attaquant vole l'identité de quelqu'un, par exemple prétendre être root (super administrateur sous linux) ;
- **La trappe** qui consiste à exploiter une faille dans un logiciel. C'est comme une porte dérobée (en anglais *backdoor*) dissimulée dans un logiciel, pour une exploitation ultérieure, par son constructeur ou par un pirate qui en a connaissance ;
- **Les buffers overflows** ou dépassement de tampon, sont une catégorie particulière de bug. Une erreur de programmation, peut entraîner l'exploitation d'un shellcode[2] à distance. Ce shellcode permettra à une personne mal intentionnée d'exécuter des commandes sur le système distant, pouvant aller jusqu'à sa destruction.

Ainsi pour se protéger contre tous ces menaces des techniques de protection sont créées telles que la cryptographie [1.10].

[2] Un shellcode est une chaîne de caractères qui représente un code binaire exécutable capable de lancer un shell ('/bin/sh' sous Unix ou command.com sous DOS et Microsoft Windows par exemple). Un shellcode peut être utilisé par un cracker voulant avoir accès à la ligne de commande.

I.5 La cryptographie

La cryptographie est une science mathématique permettant d'effectuer des opérations sur un texte intelligible afin d'assurer une ou plusieurs propriétés de la sécurité de l'information, et le rendre inintelligible à tout utilisateur non autorisé. La plupart des méthodes de chiffrement reposent essentiellement sur les deux principes que sont la substitution et la transposition.

- **La substitution** est le fait de remplacer les lettres composant le message par d'autres, ou par des symboles ;
- **La transposition** consiste à permuter les lettres du message afin de le rendre inintelligible.

I.5.1 Les clés symétriques

On utilise des clés identiques à la fois pour le chiffrement et pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète. Cette approche va entraîner un problème de gestion des clés [1.10] non négligeable. En effet il faut résoudre les questions suivantes : Qui doit connaître le secret ? Comment transmettre le secret ?

Le chiffrement symétrique est aussi appelé cryptographie symétrique ou classique. On distingue plusieurs méthodes utilisées en cryptographie classique : (DES, IDEA, AES, etc.) ???

- **Le D.E.S (Data Encryption Standard) ou Standard de Chiffrement de Données**, proposé par IBM et devenu un standard mondial depuis plusieurs années. Il a était accepté par l'administration Américaine depuis le 23 novembre 1976.

C'est un système de chiffrement par blocs qui permet de crypter 64 bits de données en utilisant une clé de 64 bits c'est-à-dire 8 caractères mais seulement 56 bits sont utilisés. L'algorithme est assez simple puisqu'il ne combine en fait que des permutations et des substitutions [1.1].

I.5.2 Les clés asymétriques

On utilise des clés différentes pour le chiffrement et pour le déchiffrement. Le système cryptographique repose sur la notion de bi-clé, c'est-à-dire un couple de variables cryptographiques :

une clé publique qui peut être publiée (mais qui doit rester intègre), et une clé privée qui doit rester secrète et dont seul le propriétaire a la connaissance et l'usage. L'utilisation de bi-clé entraîne sans doute la notion d'infrastructure de gestion de clés, afin de certifier et de gérer les clés publiques [1.4]. Il existe plusieurs algorithmes qui reposent sur la cryptographie à clé publique :

- **Le RSA** (Ron Rivest, Adi Shamir et Len Adleman 1978) : Ce système est considéré comme l'un des plus sûrs [1.7, 1.10]. RSA est très utilisé dans le commerce électronique (sécuriser les sites Web, chiffrer les emails..) et dans le réseau financier (carte visa) ;
- **Le chiffre El-Gamal** (1985) repose sur Diffie-Hellman (basé sur le logarithme discret) [1.11].

Les services assurés pour la sécurité des réseaux (par la cryptographie) sont la confidentialité, l'authentification, l'intégrité, le contrôle d'accès et la non répudiation [1.10]. ???

La Confidentialité a été définie par l'Organisation Internationale de normalisation (ISO) comme « le fait de s'assurer que l'information n'est accessible qu'à ceux dont l'accès est autorisé », c'est-à-dire la propriété qu'une information n'est ni disponible ni divulguée aux personnes, entités ou processus non autorisés. Elle assure la protection des données contre une lecture et un dévoilement des données.

L'authentification consiste à s'assurer que l'interlocuteur est bien l'entité qu'il prétend être.

L'intégrité est le fait de garantir et d'assurer que les données reçues sont exactement les mêmes que celles envoyées par une source autorisée, c'est-à-dire les données n'ont pas été supprimées, tronquées, augmentées, substituées. Si elles doivent l'être il faut s'assurer que cette modification est légitime.

Le contrôle d'accès sert à prévenir l'utilisation non autorisée d'une ressource, c'est-à-dire définir qui peut accéder à une ressource, dans quelles conditions et pour quoi faire.

La non répudiation consiste à se protéger contre un négation par une des entités d'avoir participé à toute ou une partie de la communication. On distingue la non répudiation à la source et la non répudiation à la destination.

I.5.3 Sécurisation des échanges

Puisque l'Internet repose sur IP, dont le but au début n'a jamais été d'assurer des communications sécurisées, il est donc urgent d'implémenter de nouveaux protocoles pour sécuriser les échanges. D'où la définition d'un nouveau protocole comme IPv6 (adressage sur 128 bits) qui implémente par défaut des services de sécurité contrairement à IPv4 (adressage sur 32 bits). Comme plusieurs réseaux tournent encore sur IPv4 ajouter des modules de sécurité est devenu alors une priorité avant la transition complète sur IPv6, qui sans doute ne sera pas aujourd'hui. C'est ainsi que de nouvelles applications et de nouveaux matériels ont vu le jour dont : IPSec, les routeurs filtrants, les Firewalls, les serveurs Proxy, etc ….

I.5.3.1 IPSec

IPSec (IP Security Protocol) est une norme développée depuis 1992 par un l'IETF (Internet Engineering Task Force). Aussitôt en 1995 une première version est sortie sous forme de RFC (Request For Comment) et une seconde version en 1998 qui offre plus de fonctionnalités. IPSec intervient sur les données pour garantir la sécurité. Il s'insère directement dans la pile de protocole TCP/IP plus exactement au niveau IP.

Les services offerts par IPSec sont la confidentialité, l'authenticité des données et la protection contre le réjeu. Ces services de sécurité sont rendus possibles par deux extensions du protocole IP : AH et ESP. Ces deux mécanises peuvent être utilisés séparément ou combinés. Pour obtenir tous les services de sécurité qu'offre IPSec indépendamment des deux extensions (AH/ESP) deux modes sont possibles : le mode **tunnel** et le mode **transport** [1.2].

I.5.3.2 Les fonctions de Hachage

Toujours pour sécuriser les données d'autres techniques mathématiques sont utilisées comme les fonctions de hachage. Elles convertissent un grand ensemble à un petit ensemble, c'est-à-dire si nous avons une chaîne binaire **x** de taille quelconque **t, y=H(x)** devient une autre chaîne binaire de taille fixe **n** (avec **t > n, x** est appelé pré image de **y**, et **y** est appelé empreinte de **x**). On parle de collision entre **x et x'** lorsque **H(x) =H (x')** quand **x ≠ x'**.

Une bonne fonction de hachage doit donc être:

- ✓ Facilement calculable ;

- ✓ A sens unique : c'est-à-dire impossible de trouver **x** à partir de **y** ;
- ✓ Sans collision.

Les fonctions de hachages jouent le plus souvent le rôle de signature numérique et assurent l'authentification et la non répudiation. Les algorithmes de hachage les plus utilisés sont le SHA et le MD5 :

- ➢ Le SHA (Secure Hash Algorithm) permet de calculer une empreinte de 160 bits;
- ➢ Le MD5 permet de calculer une empreinte de 128 bits.

Conclusion

Après avoir fait un tour d'horizon sur quelques notions liées à la sécurisation des systèmes et des données, nous pouvons dire que la sécurisation n'est donc pas une mission facile à remplir. Elle doit être prise par son aspect le plus général d'où la nécessité d'intégrer plusieurs outils d'ordre cryptographique, protocolaire, organisationnel, physique, architectural, système, etc. Une mise à jour quotidienne des bases antivirus des ordinateurs du système doit aussi être respectée, ainsi qu'une connaissance des nouvelles vulnérabilités des logiciels que l'on utilise dans le système d'informations.

Il ne faut pas aussi oublier le prix à payer car la sécurité a un coût. En d'autres termes elle n'est pas gratuite. Cela dépendra du niveau de sécurité. En plus de cela il faut que les administrateurs réseaux et systèmes prennent le temps d'analyser les fichiers journaux le plus souvent.

Deuxième partie
Les systèmes de détection d’intrusion

Deuxième partie : Etude des Système de Détection d'Intrusion

II.1. Définition et historique des IDS

Un Système de Détection d'Intrusion est un concept introduit pour la première fois par **JAMES ANDERSON** en 1980 [2.1]. Mais c'est la publication d'un modèle de détection des intrusions par Denning en 1987 [2.2] qui a véritablement marqué ce domaine. Aujourd'hui, les IDS font partie intégrante de la panoplie de l'administrateur de sécurité en entreprise [2.3].

Un IDS est un système qui essaye d'identifier les tentatives d'intrusion, aussi bien de la part des utilisateurs non autorisés (externes) que des utilisateurs autorisés (internes), qui abusent de leurs privilèges. C'est-à-dire qu'il permet d'observer et d'analyser l'activité sur un hôte donné ou sur un réseau. En d'autres termes il permet de voir si la sécurité d'un système est menacée en vue de prendre immédiatement les mesures de protection qui s'imposent. Plusieurs critères ont été définis pour classifier les IDS. Selon les travaux publiés par le laboratoire d'IBM de Zurich [2.7], cinq critères ont été définis :

- le principe de détection utilisé ;
- le comportement en cas d'alerte ;
- la source des données à analyser ;
- la méthode d'analyse ;
- la fréquence d'utilisation.

Pour le principe de détection utilisé, il existe deux grandes familles de méthodes de détection d'intrusion. La première famille procède par analyse comportementale (anomaly detection), et la seconde famille se base sur l'analyse par scénario (misuse detection).

II.2. Approche Comportementale

Cette approche se base sur le comportement normal de l'utilisateur et sur le fait que toute déviation par rapport à celui-ci est potentiellement suspecte. Il s'agit de dresser un profil de l'utilisateur à partir de ses habitudes, et de déclencher une alerte lorsque des événements hors profil se produisent. Au préalable, l'IDS doit alors apprendre le comportement du réseau pour établir le profil, ce qui va sans doute entraîner beaucoup de faux positifs et de faux négatifs [2.4]. Le

comportement, ou profil, d'un utilisateur ou d'une application peut être construit selon plusieurs méthodes, parmi lesquelles la méthode statistique et les réseaux de neurones.

Dans méthode statistique le profil repose sur l'hypothèse que le comportement « normal » d'un utilisateur peut être modélisé par une loi statistique. Il peut être calculé à partir de variables considérées comme aléatoires et échantillonnées à intervalles réguliers. Ces variables peuvent être :

- le temps processeur utilisé ;
- l'heure et la durée des connexions ;
- les programmes exécutés.

Le but est de calculer le taux de déviation entre les comportements passés et le comportement courant.

Les réseaux de neurones sont à l'origine d'une tentative de modélisation mathématique du cerveau humain[3]. La technique consiste à apprendre à un réseau de neurones le comportement normal d'un utilisateur. Par la suite, lorsqu'on lui fournira les actions courantes, il devra décider de leur normalité ou non [2.5].

L'approche comportementale a ses forces (détection de nouvelles attaques inconnues auparavant, détection des abus de privilèges des utilisateurs légitimes du système) et ses faiblesses (fausses alertes (faux positifs), Risque de faux négatifs si les attaques ont été commises lors de l'apprentissage).

II.3. Approche par Scénario

La plupart des IDS utilisent ce type d'approche qui fonctionne selon le principe : le comportement actuel de l'utilisateur correspond t-il à un comportement "intrusif" connu ?
Dans cette approche, des scénarios ou règles d'attaque sont construits, et l'analyse des traces d'audit se fait à la recherche de ces scénarios, en s'appuyant sur des intrusions passées ou des faiblesses théoriques connues. Une règle peut être relative à un seul événement malveillant, une séquence d'événements représentant un scénario d'intrusion. Donc l'attaque, pour être détectable, doit être décrite par un scénario.

[3] http://www.chimique.usherbrooke.ca/cours/gch445/neurones-intro.html

II.3.1 Le Système expert

Le système expert comporte une base de règles qui décrit les attaques. Les événements d'audit sont traduits en des faits qui ont une signification sémantique pour les systèmes experts. Il décrit statistiquement, grâce à sa base de données de règles, le profil d'un utilisateur par rapport à ses comportements antérieurs. Le principe ici est de comparer les règles déjà établies au comportement courant. La base des règles peut être mise à jour pour détecter de nouvelles attaques [2.5]. C'est une méthode qui n'est plus utilisée par les IDS récents.

II.3.2 Pattern Matching (recherche de motifs)

Des signatures d'attaques sont fournies, à des niveaux sémantiques divers selon les outils (de la suite d'appels système aux commandes passées par l'utilisateur). C'est la méthode la plus utilisée actuellement et la plus facile à comprendre. ??? D'autres méthodes telles que les algorithmes génétiques ont été proposées par JOHN HOLLAND dans les années 70 [2.6] et les réseaux de neurones [2.5].

L'approche par scénario présente l'avantage de la prise en compte des comportements exacts des attaquants potentiels. Elle présente aussi l'inconvénient de ne détecter que les attaques décrites dans la base des règles, correspondant aux attaques connues.

On peut aussi classifier les IDS selon le comportement à adopter après qu'une alerte ait été remontée, c'est-à-dire les mesures à prendre après la détection d'une menace sérieuse. Certains IDS se contentent de remonter seulement des alertes (réponses passives), d'autres prennent des mesures « correctives » (réponses actives). Un IDS peut être configuré avec certaines entités (routeurs, Firewall, proxy …etc.) pour bloquer ou stopper les menaces en cours. Comme réponse active il est possible de :

- Couper les connexions suspectées (envois d'un "*ResetKill*") c'est-à-dire construire un paquet TCP FIN pour forcer la fin d'une connexion ; ceci est uniquement valable sur des techniques d'intrusion utilisant le protocole de transport TCP;
- Bloquer certains ports ou reconfigurer le Firewall ;
- Démarrer une application ou un programme extérieur pour exécuter une action spécifique telle que : envoyer un message sms ou un fax, émettre une alerte auditive, composer un numéro de téléphone.

Les IDS peuvent aussi être configuré pour émettre des réponses passives. Comme réponse passive il est possible de :

- Envoyer un e-mail à un ou plusieurs utilisateurs pour notifier d'une intrusion sérieuse ;
- Sauvegarder dans une base de données centrale des détails de l'alerte comme par exemple le timestamp, l'adresse IP de l'intrus, l'adresse IP de la cible, le nom du protocole utilisé, la charge utile (payload).

Les IDS peuvent aussi être classifiés selon la source de donnée à analyser. Nous pouvons avoir plusieurs sources de données provenant :

- des fichiers générés par les systèmes d'exploitation ;
- des fichiers générés par les applications ;
- des paquets obtenus en écoutant le trafic réseau.

Les deux premiers sont utilisés dans les systèmes de détection basés sur le host : H-IDS[4], le troisième est utilisé dans les systèmes de détection basés sur le réseau : N-IDS[5]

II.4. La détection d'Intrusion basée sur l'hôte

Les HIDS sont des systèmes de détection d'intrusions basés sur un hôte. La surveillance de l'HIDS se résume à un seul hôte. Les sources de données qu'ils utilisent proviennent généralement des traces d'audit système, des historiques des commandes à exécuter ou des appels système évoqués par les applications. Le HIDS utilise aussi les paquets réseaux entrants/sortants de l'hôte pour y déceler des signaux d'intrusions (Déni de Services, Backdoors, chevaux de troie, tentatives d'accès non autorisés, exécution de codes malicieux, attaques par débordement de buffers…). Les HIDS sont en général placés sur des machines sensibles, susceptibles de subir des attaques et possédant des données sensibles pour l'entreprise. Les serveurs web, ftp et applicatifs, peuvent notamment être protégés par un HIDS. Ils présentent des avantages ainsi que des inconvénients répertoriés dans le tableau 1.

[4] HIDS : Host-based Intrusion Detection Systems

[5] N-IDS : Network-based Intrusion Detection Systems

Tableau 1 : Avantages et inconvénients des HIDS

Avantages	Inconvénients
indépendant de la topologie du réseau; donne l'information sur le succès ou l'échec d'une tentative de connexion ; découvre plus facilement un Cheval de Troie[6] puisque les informations et les possibilités sont très étendues ; détecte des attaques impossibles à détecter avec des IDS réseau puisque le trafic y est souvent crypté ; accède à des composants non accessibles sur le réseau (la base de registre de Windows) ; permet d'observer les activités sur l'hôte avec précision.	difficiles à gérer puisse que chaque équipement doit être configuré individuellement ; n'offrent pas une vision d'ensemble de l'attaque si plusieurs équipements d'un réseau sont attaqués ; sont plus vulnérables aux attaques de type DoS ; utilisent beaucoup de ressources de l'équipement hôte (occupent beaucoup d'espace disque, consomment trop de ressources CPU).

Beaucoup d'HIDS ont été implémentés. Certains sont du domaine public, d'autres sont commerciaux. On distingue :

- Tripwire téléchargeable sur : http://www.tripwire.com/products/index.cfml;
- DragonSquire téléchargeable sur : http://www.enterasys.com/ids/squire/;
- Tiger téléchargeable sur http://freshmeat.net/redir/tiger-audit/30581/url_homepage/tiger.

II.5. Détection d'intrusion basée sur le réseau (NIDS)

Les NIDS (Network Based Intrusion Detection Systems) sont des systèmes de détection d'intrusion basés sur le réseau. Leur rôle principal est d'analyser et d'interpréter le trafic réseau en vue de trouver des activités suspicieuses Pour cela ils collectent et analysent les paquets de

[6] Cheval de Troie : un programme normal qui héberge, cache un notre programme malveillant

données circulant sur le réseau grâce aux capteurs. Ces derniers sont le plus souvent des hôtes placés un peu partout sur le réseau en vue de remonter des alertes si nécessaire. Chaque paquet qui arrive sera analysé pour en déduire 5 propriétés :

- P1 : Le paquet contient une signature d'attaque ;
- P2 : Le paquet est un sous mot d'une signature d'attaque ;
- P3 : Le paquet contient un suffixe qui est un préfixe d'une signature d'attaque ;
- P4 : Le paquet contient un préfixe qui est un suffixe d'une signature d'attaque ;
- P5 : Le paquet ne comporte aucune partie d'une signature d'attaque.

Certains IDS basent leur analyse sur la corrélation des données provenant aussi bien des fichiers d'audit d'un ou plusieurs hôtes, que du trafic réseau, afin de détecter le plus d'intrusions possibles. Dans cette situation, c'est le " stealth mode" (mode furtif) qui est utilisé. Ce dernier rend invisible le capteur par rapport aux autres machines du réseau. Ainsi il sera très difficile à détecter et par conséquent à attaquer. Les NIDS sont plus utilisés que les HIDS, et ils jouent un rôle très important dans la politique de sécurité des réseaux et systèmes informatique. Cependant ils présentent néanmoins beaucoup d'insuffisances comme le montré le tableau 2.

Tableau 2 : Avantages et inconvénients des NIDS

Avantages	Inconvénients
indépendants des systèmes d'exploitation installés sur tout le réseau surtout sur les équipements du réseau ; surveillent tout le trafic réseau sur des services connus (ex. http, port 80) ; donnent des informations fiables, même après la compromission d'un ou plusieurs équipements sur le réseau; peuvent être bien sécurisés puisqu'ils se "contentent" d'observer le trafic ; détectent facilement des attaques distribuées et les scan de réseau.	des vulnérabilités à certaines attaques ; une baisse de performance à des heures de trafic soutenu (les débits réseau augmentent de manière exponentielle car les vitesses de transfert sont décuplées tous les deux ans [2.14]) ou dans le cas de congestion du réseau; impossibilité de dire si une attaque a réussi ou non. En effet la plupart des NIDS donnent l'information qu'une attaque a été initiée mais ne donnent pas l'impact d'une attaque.

Il existe plusieurs NIDS commerciaux ou du domaine public

- Snort téléchargeable à partir du site www.snort.org;
- ISS de RealSecure téléchargeable sur http://www.securisa.com/p_doc0506231.html;
- Dragon téléchargeable à partir de http://www.securitywizards.com ;
- NetRanger disponible à partir de http://www.cisco.com ;
- Bro disponible à partir de http://www.bro-ids.org.

II.6. Les IDS basés sur une application (ABIDS)

Les IDS basés sur les applications sont un sous-groupe des IDS hôtes. Ils contrôlent l'interaction entre un utilisateur et un programme en ajoutant des fichiers de log afin de fournir de plus amples informations sur les activités d'une application particulière. Puisque nous opérons entre un utilisateur et un programme, il est facile de filtrer tout comportement notable. Un ABIDS se situe au niveau de la communication entre un utilisateur et l'application surveillée. L'avantage de cet IDS est qu'il lui est possible de détecter et d'empêcher des commandes particulières dont l'utilisateur pourrait se servir avec le programme, et de surveiller chaque transaction entre l'utilisateur et l'application. De plus, les données sont décodées dans un contexte connu, leur analyse est donc plus fine et précise.

Par contre, du fait que cet IDS n'agit pas au niveau du noyau, la sécurité assurée est plus faible, notamment en ce qui concerne les attaques de type "Cheval de Troie". De plus, les fichiers de log générés par ce type d'IDS sont des cibles faciles pour les attaquants et ne sont pas aussi sûrs, par exemple, que les traces d'audit du système [2.10]. Ce type d'IDS est utile pour surveiller l'activité d'une application très sensible, mais son utilisation s'effectue en général en association avec un HIDS. Il faudra dans ce cas contrôler le taux d'utilisation CPU des IDS afin de ne pas compromettre les performances de la machine.

Parmi les autres types systèmes de détections on peut noter : les Multihost-based IDS et les IDS hybrides.

Les **Multihost-based** IDS sont chargés de surveiller plusieurs hôtes et utilisent des informations provenant des fichiers d'audit de ces hôtes comme source principale d'entrées pour détecter les intrusions. Dans ce cas c'est la corrélation des données d'audit des différents hôtes qui permettra de détecter des comportements intrusifs. Cette analyse est faite le plus souvent sur un hôte séparé. Cela donc va poser pas mal de problèmes, liés à la machine qui fait les traitements

(puissance du processeur, mémoire vive, espace disque …), et à la nature même du SE installé sur la sonde, si les données à analyser proviennent des machines ayant des SE différents.

Les IDS **hybrides** sont une famille d'IDS née pour regrouper entre autres les avantages des IDS comportementaux et des IDS par scénario. Prélude est un exemple d'IDS hybride.

Les IDS peuvent aussi être classifiés par rapport à la manière dont les données collectées seront analysées. Cette analyse des données peut être centralisée ou distribuée. On peut aussi se baser sur l'utilisation de l'IDS qui peut être continue ou périodique. De même le comportement à adopter après une détection d'attaque (réaction passive ou active) peut aussi caractériser un IDS.

Les caractéristiques d'un IDS peuvent être résumées sur la figure 2.1.

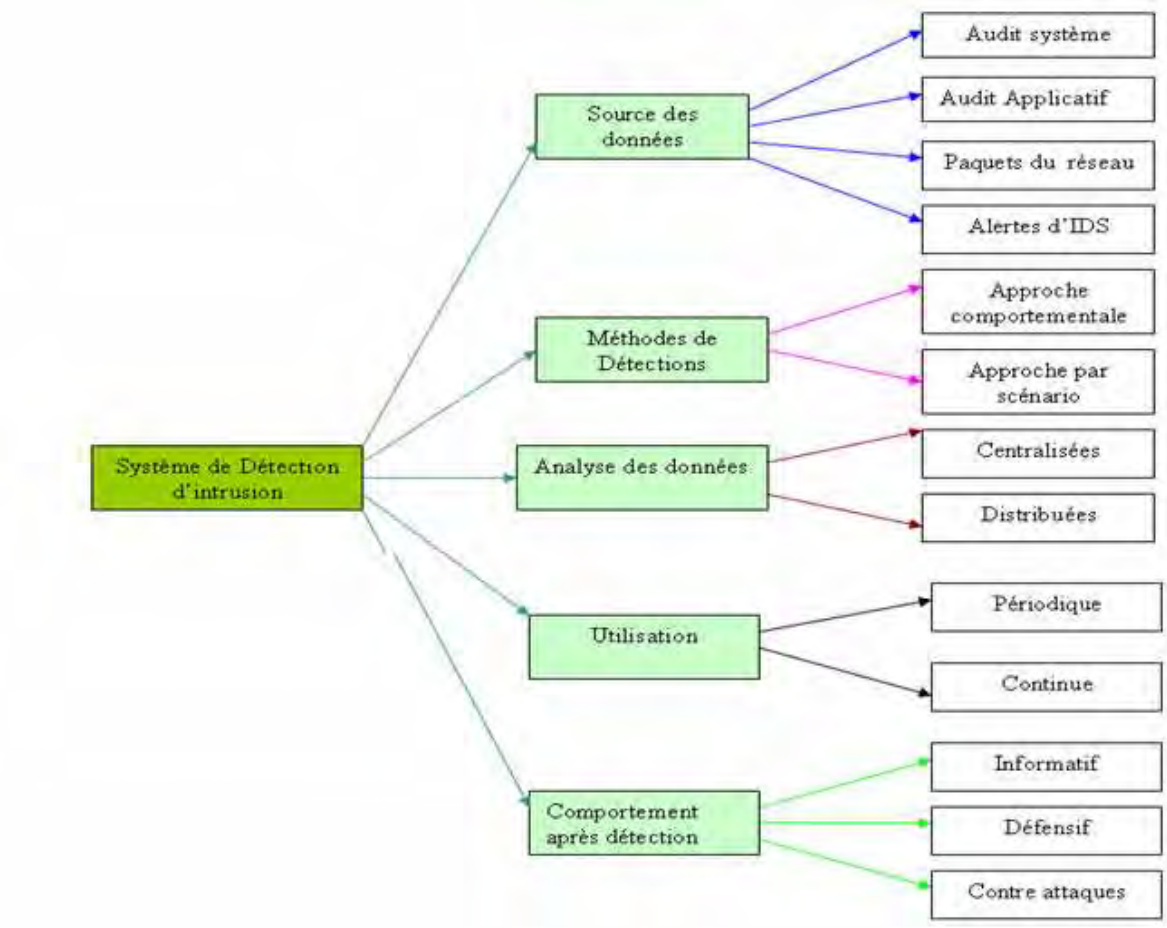


Figure 2. 1: Classification générale des IDS

II.7. Architecture de base d'un IDS

Un IDS est constitué généralement de trois composants : un capteur, un analyseur et un manager.

Le capteur se charge de la collecte d'informations. Il suit l'activité du système en se basant sur plusieurs sources de données pour fournir à l'analyseur une séquence d'événements, qui renseignent sur l'état du système. Ces données peuvent être transmises en brut, mais le plus souvent un prétraitement est effectué. Ce traitement permet de diminuer la quantité de données à transmettre. On distingue classiquement trois types de capteurs en fonction des sources de données utilisées :

- **Les capteurs système** collectent des données produites par les SE des machines, notamment par le biais des journaux d'audit système, ou par celui des appels systèmes invoqués par les applications. Ces types de capteurs sont utilisés dans les HIDS ;
- **Les capteurs réseau** collectent les données en écoutant le trafic entre les machines, par le biais d'une interface spécifique. On les utilise dans les NIDS ;
- **Les capteurs applicatifs** collectent les données produites par une application particulière, avec laquelle des utilisateurs sont susceptibles d'interagir. Comme application on peut avoir un serveur Web ou un serveur de base de données.

Le moteur d'analyse a pour rôle de voir si le flux d'événement qu'il a reçu du capteur contient des éléments caractéristiques d'une activité malveillante. Cette comparaison repose sur les deux principes de détection utilisés dans les IDS que nous avons évoqués plus haut que sont l'approche comportementale et l'approche par scénario.

Le module de sortie se charge de réaliser les instructions adéquates lorsqu'une règle de détection est vérifiée. Il permet d'imprimer un message d'alerte à l'écran, d'écrire dans un fichier *log*, ou d'envoyer un message à un serveur distant (centralisation de données).

L'architecture d'un IDS peut être représenté par la figure 2.2 :

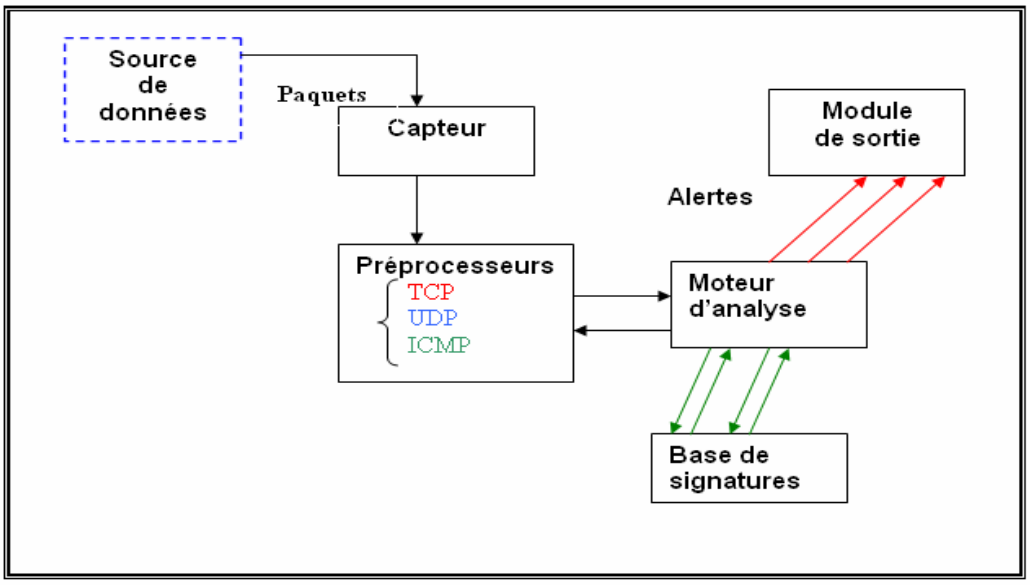


Figure 2. 2 : Architecture de base d'un NIDS

II.8 Etude de quelques IDS Réseau

II.8.1 Snort

Snort est un logiciel open source appartenant à la famille des IDS. Il a été développé en 1998 par Marty Roesh et est disponible sous licence GNU. Son code source est accessible et modifiable à partir du site web de Snort[7]. C'est un NIDS[8], qui observe et analyse le trafic réseau, en vue de trouver des indicateurs d'attaque afin de déclencher des alertes.

Au début Snort n'était qu'un simple outil de gestion réseau et est devenu maintenant la sonde la plus puissante et la plus répandu soit plus 100 000 de déploiement dans le monde [3.2]. C'est un outil très flexible, qui peut être installé sur plusieurs plateformes :

- Processeurs: i386, Sparc, PowerPC, Alpha;
- OS: Linux, OpenBSD, FreeBSD, Solaris, HP-UX, AIX, Mac OS X,Windows (win9x /NT/2000/XP).

Il peut fonctionner avec plusieurs types de dépôt de données tels que : MySQL, Oracle, MS SQL, fichier plat, XML, syslog, etc … Snort peut être configuré pour fonctionner en quatre modes : mode sniffer, mode enregistreur de paquets et mode de détection d'intrusion.

[7] http://www.snort.org
[8] NIDS (Network Intrusion Detection System)

Le mode sniffer « renifleur » : Dans ce mode, Snort lit les paquets circulant sur le réseau et les affiche d'une façon continue sur l'écran ; il fonctionne ici exactement comme *tcpdump* (sniffeur de paquets) mais avec beaucoup plus de fonctionnalités [3.3]. Nous pouvons voir les entêtes ainsi que les charges des paquets en spécifiant les options :

- ✓ **Snort -v** : pour afficher les entêtes des paquets TCP/IP ;
- ✓ **Snort -vd** : affiche tout le paquet TCP/IP (entêtes et données) ;
- ✓ **Snort –vde** : affiche tout le paquet TCP/IP (entête et données) ainsi que l'entête de la couche de liaison de données.

Le mode « packet logger » ou mode enregistreur de paquets consiste à journaliser le trafic réseau dans des répertoires sur le disque sous un format spécifique. L'enregistrement du trafic sous un format donné permet son exploitation par des outils comme Ethereal, tcpdump etc.…

Le mode détection d'intrusion est la fonctionnalité de SNORT la plus intéressante c'est-à-dire en mode IDS. Ici SNORT est lancé avec un fichier de configuration avec l'option –c. Dans ce mode SNORT analyse le trafic réseau en le comparant avec des règles déjà définies. Selon sa configuration, il établit les actions à exécuter. Il existe plus de 2550 règles [3.3]. On peut aussi créer ses propres règles. Quand une alerte est déclenchée, les paquets sont journalisés automatiquement vers le mécanisme d'enregistrement qui peut être une base de données ou un répertoire par défaut */var/log/Snort*. Les alertes peuvent également être envoyées à Syslog, ou elles peuvent être envoyées comme des messages *WinPopup* avec *smbclient* ou encore par email.

Comme exemple de règle nous pouvons avoir :

Alert tcp any any -> 192.168.1.0/24 80 (flags: A ;\content : "passwd"; msg: "detection de `passwd' " ;)

Cette règle permet de générer un message d'alerte "détection de passwd" lorsque le trafic, à destination d'une machine du réseau local 192.168.1.0/24 vers le port 80, contient la chaîne « passwd » (spécifiée par l'utilisation du mot-clé « content »), et que le flag ACK du header TCP est activé (flags : A).

Le mode « prévention des intrusions réseau » : le trafic est bloqué en cas de découverte d'actions suspicieuses. Mais cela pose un certain nombre de problèmes en cas de fausse alerte, on peut avoir un déni de service.

SNORT est donc un outil complet allant de l'écoute du réseau jusqu'à l'arrêt du trafic jugé malveillant. Ainsi pour la recherche des motifs dans ces derniers SNORT utilise des algorithmes de

fouille de données. Avant d'évoluer vers l'algorithme de Wu-Manber, SNORT utilisait l'algorithme d'Aho-Corasick qui sont des algorithmes de recherche mots dans un texte [3.4].
Il présente des avantages et des inconvénients [3.2] :

Tableau 3: Avantages et inconvénients de SNORT

	Avantages	**Inconvénients**
Snort	– logiciel libre (Open Source) – multi-plateforme – personnalisation les règles – fonctionnement en plusieurs modes – possibilité de stocker les alertes dans une base de donnés – analyse en temps réel	– fichiers d'alertes difficiles à parser de manière automatisée – configuration essentiellement par édition de fichiers texte – beaucoup d'autres fonctionnalités sont commerciales – Faux positifs et négatifs

II.8.2 RealSecure de ISS

RealSecure est la solution IDS proposée par la société ISS (Internet Security Systems). Il surveille le trafic réseau, par le biais d'un agent, dans le but de découvrir des signatures d'attaques. Ces dernières sont élaborées par le centre de veille d'ISS, nommé la X-Force. Cette solution s'appuie assez naturellement sur une architecture du type de celle proposée par l'IDWG [3.6].

En effet l'architecture IDWG contient des capteurs qui envoient des évènements à un analyseur. Un ou plusieurs capteurs couplés avec un analyseur forment une sonde. Une sonde envoie des alertes vers un manager qui la notifie à un opérateur humain.

La gamme RealSecure propose RealSecure Network Sensor et RealSecure Server Sensor. RealSecure Network Sensor s'installe sur une station dédiée pour contrôler le trafic réseau à la recherche de signatures d'attaques. RealSecure Server Sensor protège les serveurs stratégiques par l'analyse des événements dans le noyau du SE, des journaux de connexions et de l'activité réseau de ces serveurs. Il peut être installé sur différentes plateformes matérielles et supporte plusieurs

types de SE (Windows, Linux, Solaris) [3.7]. Comme tous les IDS, RealSecure génère aussi des faux positifs et négatifs.

II.8.3 Bro

C'est un outil permettant de détecter les intrusions en temps réel, et est destiné aux réseaux Gbps à fort trafic [3.8]. Il est aussi un logiciel open source et surveille passivement le trafic réseau à la recherche d'activités soupçonneuses, définies dans une base de signatures à cette effet.

Bro s'installe uniquement sur les OS de la famille Linux/Unix (Linux, FreeBSD, Solaris, OpenBSD). Il est l'un des meilleurs en matière d'analyseur protocolaire. En effet, il est important de noter que Bro a été développé principalement comme plate-forme de recherche pour la détection d'intrusion et l'analyse de trafic [3.9]. Son architecture de base est représentée par la figure 2.3.

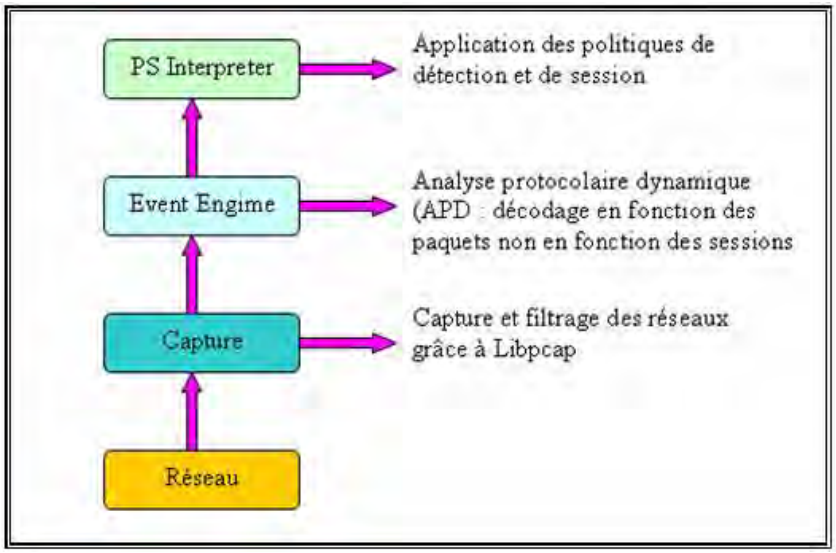


Figure 2. 3: Architecture de l'IDS Bro

Bro présente des d'avantages mais aussi des insuffisances, parmi lesquels on distingue :

- C'est un logiciel à code ouvert;
- L'analyse des paquets se fait en temps réel ;
- Plus adapté pour les réseaux à Gbits ;
- S'installe uniquement sur les plateforme Linux/Unix.

II.8.4 Prélude

Prélude est un IDS un hybride, c'est-à-dire qu'il est à la fois un HIDS et un NIDS, et même plus encore [3.10, 3.11]. C'est un IDS distribué pour les intrusion et les anomalies sous licence GPL. Son aarchitecture modulaire et naturellement distribuée (sondes et managers) permet de n'installer que ce qu'il faut là où il faut, sans surcharger le réseau ni les systèmes hôtes. Grâce à des sondes dédiées, il autorise la prise en compte, au sein d'un schéma unique et homogène, de la quasi-totalité des évènements de sécurité, que cela soit au niveau réseau ou local.

La surveillance est réalisée par l'analyse du trafic réseau et l'utilisation de signatures d'évènements hostiles (NIDS) ou par la scrutation continue de différents fichiers de *logs* (HIDS). Les sondes, réseau comme locales, n'effectuent que les opérations de surveillance et de génération des alertes, alors que les managers prennent en charge le stockage ou la réexpédition de ces dernières. Les différents composants de Prélude sont [3.10, 3.11]:

- Libprelude (la librairie Prélude) ;
- Prélude-Nids (la sonde réseau) ;
- Prélude-LML (la sonde locale) ;
- Prélude-Manager (le contrôleur) ;
- Prélude-Frontend ou Prewikka (l'interface web).

L'architecture de prélude est présentée par la figure 2.4.

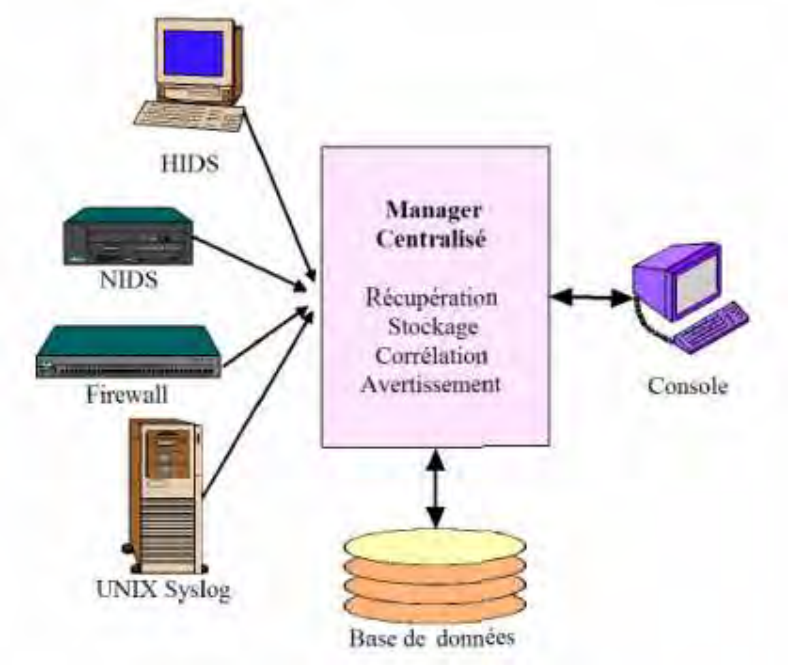


Figure 2. 4: Architecture de prélude

C'est au niveau de **Prélude-Manager** que toutes les alertes seront regroupées et enfin un mécanisme de corrélation d'alertes sera mis en place pour diminuer les faux positifs. Ainsi, l'administrateur de sécurité aura une vue globale sur les attaques qui ont réellement abouti, sans devoir visualiser les unes après les autres les milliers d'alertes générées. Cette partie a donc fait l'objet d'une implémentation PHP d'une interface Web (offrant une fonctionnalité de corrélation) pour prélude.

Quelques avantages et inconvénients de Prélude :

- un système de détection hybride ;
- modulaire et architecture client serveur ;
- nombre d'alertes remontées plus important que Snort ;
- La grande force de Prelude-IDS est de pouvoir intégrer les fonctionnalités d'autres outils de sécurité de référence. On peut, par exemple, utiliser Honeyd comme une sonde, envoyer les résultats vers le manager qui les intègrera ensuite dans la base de données ;
- Prélude est très peu documenté, et la plupart des documents contiennent parfois des erreurs ;
- moins populaire que Snort ;
- ne s'installe que sur les systèmes Unix/Linux.

II.8.5 NetRanger

C'est un outil de détection d'intrusion réseau proposé par CISCO qui est appelé maintenant Cisco IDS. Ce système est composé de deux éléments majeurs que sont le capteur et la sonde.

Le capteur Secure IDS que l'on peut placer à un endroit spécifique du réseau qui doit être surveillé. Plusieurs capteurs peuvent être installés pour surveiller différents emplacements. Ce capteur détecte toute activité non autorisée sur le réseau, en analysant le trafic par rapport à des fichiers de signatures, basés sur des règles.

Lorsqu'il détecte une activité non autorisée, le capteur peut envoyer des alarmes à une console de gestion en indiquant le détail de l'activité. Il peut aussi contrôler d'autres systèmes, tels que des routeurs, pour interrompre la session non autorisée.

La console Secure IDS Director est un système de gestion logicielle qui surveille de façon centralisée l'activité d'un ou de plusieurs capteurs Cisco Secure IDS situés sur des segments du réseau local ou distant. Cette console permet aux administrateurs réseau d'identifier précisément et

rapidement le lieu, et le type d'attaque, de définir le degré de gravité et d'y répondre instantanément [3.12].

L'étude de ces différents NIDS nous a permis de faire un tableau récapitulatif sur les attaques que ces derniers peuvent détecter ou non.

Tableau 4 : Récapitulation des attaques détectées en fonction des NIDS

	Scan TCP SYN	Scan furtif	buffer overflow	Backdoor	ArpSpoof du protocole ARP	TCP UDP ICMP	Attaque CGI
Snort	OUI	OUI	OUI	NON	**NON**	OUI	OUI
Bro	OUI	NA	NON	NON	NA	NA	OUI
RealSecure de ISS	OUI	NA	OUI	NA	OUI	OUI	NA
Prélude	OUI	OUI	OUI	NA	OUI	NA	OUI
NetRanger	OUI	NA	OUI	NON	NON	OUI	NA

L'analyse de ce tableau montre que les NIDS ne détecte pas toutes ces différentes attaques possibles. Pour ce qui nous concerne notre objectif était de voir le comportement des IDS par rapport aux attaques ARP. Ainsi l'attaque ArpSpoof (dont nous allons détailler le principe de fonctionnement) n'est pas détectée dans Snort. Nous verrons pourquoi ce type d'attaque n'est pas remonté par Snort. Nous essaierons de proposer un algorithme qui permettra de prendre en compte ce type d'attaque.

II.8. La Corrélation d'alerte

La corrélation d'alerte [2.11] est un terme relativement nouveau dans le domaine des IDS. La base théorique est encore en construction dans la littérature, mais on peut toutefois identifier deux approches dans la corrélation d'alerte en détection d'intrusion : la corrélation implicite et la corrélation explicite. L'objectif de la corrélation d'alerte est de réduire la quantité d'alertes que l'opérateur doit traiter, d'améliorer la qualité du diagnostic, c'est à dire augmenter l'exploitabilité des alertes déclenchées pour le système de détection d'intrusion, qui sont le plus souvent de bas niveau.

La corrélation implicite repose sur une fonction de similarité, c'est-à-dire faire une correspondance fréquentielle ou statistique entre des alertes.

La corrélation explicite repose sur un langage d'expression de scénarios où des règles sont définies. Un scénario regroupe en général un ensemble de propriétés que doivent satisfaire les alertes, et des liens les connectant [2.8].

Ces deux formes de corrélation sont à rapprocher des deux approches classiques de la détection d'intrusion : l'approche comportementale et l'approche par scénario.

II.12. Positionnement des IDS dans une architecture réseau

Pour jouer un rôle déterminant dans la politique de sécurité les IDS doivent, en plus d'être bien configurés être positionnés à des endroits spécifiques. Leurs emplacements dépendent de ce que l'on veut surveiller ou observer. Plusieurs architectures ont été proposées à différents endroits sur le réseau. Ils peuvent être placés avant ou après le pare-feu ou même dans la DMZ[9], où l'on place les serveurs visibles de l'extérieur.

II.12.1 Positionnement d'un IDS avant le pare-feu

Cet emplacement permet à l'IDS de remonter toutes les attaques venant de l'Internet. L'intérêt de cette position est que le capteur reçoit et analyse l'ensemble du trafic de l'Internet. Dans cette position nous ne sommes pas certains que toutes les attaques sont détectées et filtrées, mais nous avons l'avantage d'obtenir plus d'informations dans les fichiers log, et l'ensembles des attaques qui sont dirigées vers le pare-feu.

[9] DMZ (DeMilitarized Zone) : qui est une zone démilitarisé c'est-à-dire très surveillée

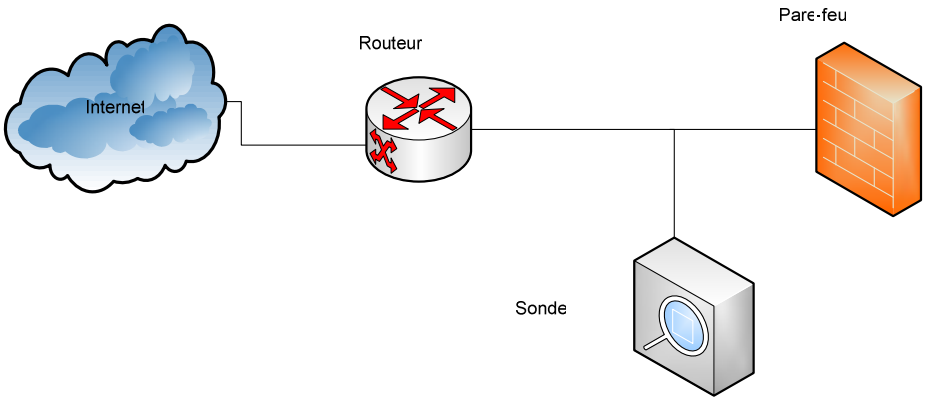


Figure 2. 5: Positionnement d'un IDS avant le pare-feu

Les capteurs placés à cet endroit servent généralement à détecter toutes les attaques frontales en direction du réseau. Leur tâche ici est donc plus de contrôler le fonctionnement et la configuration du pare-feu que d'assurer une protection contre toutes les intrusions détectées (certaines étant traitées par le pare-feu). Il est alors difficile de faire la distinction entre les attaques pertinentes (attaques pouvant aboutir) par rapport au réseau de l'entreprise de celles qui n'avaient aucune chance de réussir étant donné les règles de filtrage du pare-feu. Le positionnement de l'IDS va entraîner la création d'une grande quantité de log [2.13]. Ainsi l'administrateur du réseau doit :

- soit disposer de beaucoup de temps pour effectuer une analyse utile des traces de son IDS ;
- soit disposer d'un excellent outil permettant de réaliser ce travail à sa place, pendant qu'il s'occupe d'autre chose ;
- soit configurer plus finement les seuils de déclenchement des alertes afin de ne pas être importuné trop souvent, car des alertes trop fréquentes sont connues pour abaisser la vigilance de l'administrateur de l'IDS, qui finit par ne plus tenir compte de celles-ci.

Par ailleurs, les alertes remontées par les IDS sont souvent configurées sur une échelle de gravité prenant en compte essentiellement les attaques susceptibles de réussir sur le réseau de l'entreprise (ou dans son DMZ). Ainsi, un réseau entièrement constitué de machines Unix pourra voir son IDS configuré pour ne pas remonter d'alertes relatives aux environnements Windows. Le risque est alors d'oublier de réactiver ces alertes lorsque l'on met en place des machines Microsoft ultérieurement. Ceci peut alors entraîner une vulnérabilité de notre réseau aux multiples attaques.

II.12.2 **Positionnement d'un IDS après un pare-feu**

L'emplacement de l'IDS avant le pare-feu avait soulevé un certain nombre de problèmes. Ainsi pour diminuer un peu ces insuffisances, l'alternative consiste alors à placer l'IDS derrière le Firewall c'est-à-dire dans le réseau local de l'entreprise. Le placement d'un IDS à cet endroit possède deux avantages :

- filtrage des attaques inefficaces par le Firewall, donc meilleure exploitabilité de l'IDS ;
- détection uniquement des attaques traversant le Firewall.

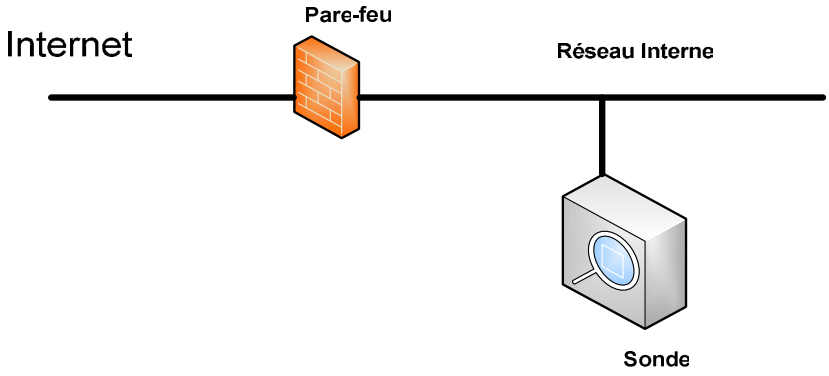


Figure 2. 6: Positionnement d'un IDS derrière le pare-feu

Ce type de positionnement masque alors complètement la menace « Internet » pour la remplacer par un indicateur de la menace résiduelle non prise en charge par les équipements actifs (filtrants) du réseau. Ce type de positionnement implique alors nécessairement la mise en place d'un plan de gestion d'incidents efficace, permettant de traiter au plus tôt les alertes remontées par l'IDS. En effet, s'agissant d'attaques a priori efficaces, car passant la barrière du Firewall, il convient de s'assurer, pour chacune d'entre elles, qu'elle n'a pas abouti, c'est-à-dire que le serveur visé n'est pas vulnérable et peut résister à ce type d'attaque. L'IDS peut ici rendre compte des attaques internes, provenant du réseau local de l'entreprise. Il peut être judicieux d'en placer un à cet endroit étant donné le fait que 80% des attaques proviennent de l'intérieur [2.13].

On peut aussi avoir une architecture qui place en même temps des sondes avant et après les Firewall ou même sur la DMZ. C'est ce que nous avons sur la figure 2.4.

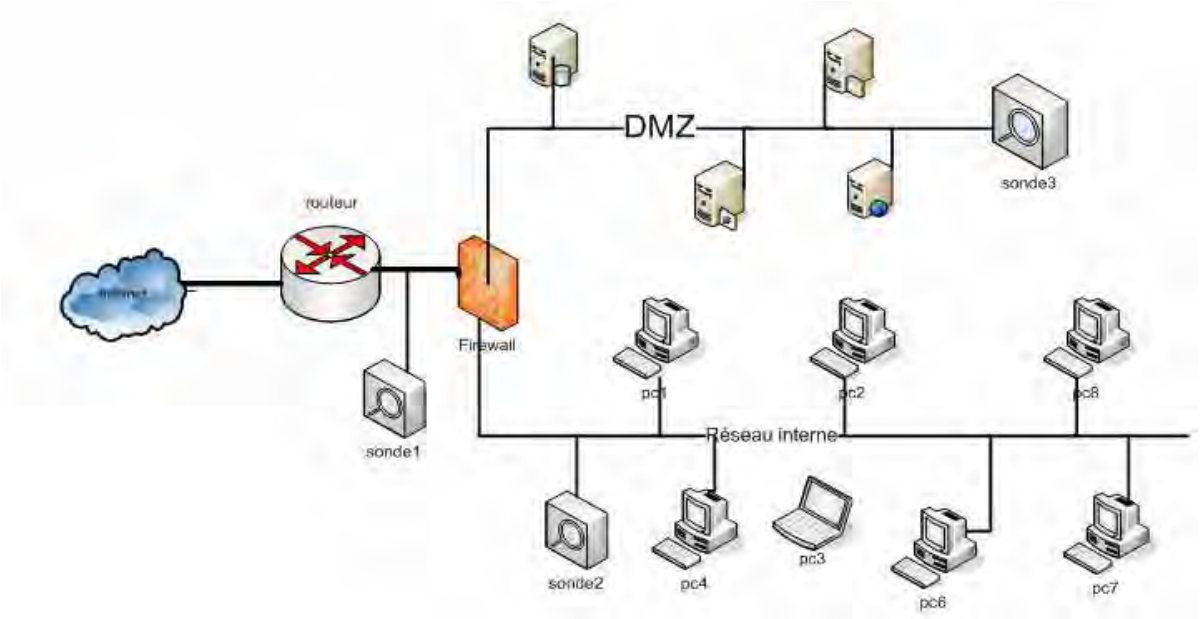


Figure 2. 7: Emplacement des IDS à différents niveaux

II.13. Méthode utilisée pour contourner les IDS

L'utilisation des IDS dans la politique de sécurité des réseaux informatiques, malgré les nombreux problèmes liés à leurs utilisations, gagne de plus en plus la confiance des administrateurs réseau. C'est pourquoi les pirates, eux aussi, profitent des limites (bases de signatures obsolètes, faux positifs, faux négatifs ...) pour contourner ces outils.

Pour illustrer ces différentes méthodes, on distingue quelques exemples d'attaque permettant d'outrepasser les IDS : méthodes classiques de scan, méthodes par le flood, méthodes par fragmentation et la méthode par noyade sous les faux positifs.

Le scan furtif SYN implémenté par NMAP est une méthode de scan classique. Nous avons étudié son principe de fonctionnement dans la première partie.

La méthode par fragmentation quant à elle repose sur la fragmentation des paquets IP pour empêcher les NIDS de détecter les attaques sachant que les paquets seront ré-assemblés au niveau du destinataire. Il est possible aussi d'envoyer des paquets IP mal fragmentés, qui vont utiliser la faiblesse de certaines piles IP (peut-être aussi celle de la sonde IDS qui capte tout le trafic) pour perturber le système.

Enfin la méthode par noyade de faux positifs repose sur la provocation de nombreuses remontées d'alertes en parallèle quand, en même temps, une attaque réelle contre le réseau est lancée. Ainsi l'administrateur, occupé à analyser les alertes, ne s'en rendra pas compte sur le moment. Nous pouvons utiliser Nmap avec certaines options pour initier cette attaque. D'autres outils tels que Nessus et Tcpdump peuvent aussi être utilisés.

Ainsi les IDS présentent des avantages et les inconvénients, ils sont présentés dans le tableau 4.

Tableau 5: Tableau récapitulatif des avantages et inconvénients des IDS

	HIDS	NIDS	IDS basé sur une application
Avantages	-sont indépendants de la topologie réseau -détectent facilement un Cheval de Troie -accèdent à des composants non accessible sur le réseau (base de registre pour Windows)	-surveillent un grand réseau et sont indépendants des systèmes d'exploitation -offrent une vision d'ensemble sur une attaque -détectent facilement des attaques distribuées	-peuvent détecter et empêcher des commandes particulières -surveillent chaque transaction entre l'utilisateur et une application
Inconvénients	-sont vulnérables aux attaques DoS -n'offrent pas une vision générale sur une attaque -utilisent beaucoup de ressource (disque, CPU…)	-ne donnent pas l'impact d'une attaque -manquent de fiabilité si le trafic réseau est soutenu -sont vulnérables à certaines attaques -ne peuvent pas analyser un trafic chiffré (VPN) -communiquent difficilement avec les commutateurs modernes	-n'agissent pas au niveau du noyau -ne détectent pas le cheval de Troie -sont le plus souvent couplés avec un HIDS

II.14. Les IPS (Intrusion Prevention System)

IPS signifie Système de Prévention des Intrusions. Le terme IPS est très nouveau. Son arrivée sur le marché de la sécurité informatique résulte de la nécessité d'améliorer, encore et toujours, les solutions existantes ayant prouvé leurs limites face aux multitudes menaces auxquelles sont confrontés les réseaux et systèmes informatiques. Son rôle est d'anticiper sur les

attaques des pirates en temps réel afin de les empêcher. Il ne s'agit plus seulement de réagir à une attaque en cours, mais d'empêcher que celle-ci puisse seulement débuter, tandis que le système de détection a le rôle d'identifier des intrusions en vue de les annoncer. Tous ces deux systèmes sont basés sur l'analyse des données qui peuvent provenir de source différentes [2.9].

Certains analystes pensent que le passage de l'IDS à l'IPS n'est qu un tour de passe-passe pour redynamiser le marché. Sur ce marché, on trouve deux acteurs phares que sont ISS et Cisco, suivent ensuite (dans le désordre) Network Associates, Snort, Symantec, NetASQ, Top Layer Networks, Netscreen, Hogwash, TippingPoint... [2.9].

II.15. Les Honeypots

Les honeypots, ou « pot de miel » en français, se définissent comme étant un système informatique public volontairement vulnérable à une ou plusieurs failles connues, visant à attirer les pirates afin d'étudier leurs stratégies d'attaque pour mieux les comprendre afin de les anticiper [2.16]. Ces comportements sont enregistrés dans un fichier journal. C'est donc une machine leurre qui collecte des informations sur les pirates.

L'arrivée des Honeypots sur le marché de la sécurité informatique remonte vers 1986 et leur mise en œuvre requiert un haut niveau d'expertise. Selon Martin Roesch, créateur de l'IDS réseau Open Source Snort et fondateur de la société Sourcefire (la version commerciale de Snort), les Honeypots sont classés en deux catégories : les Honeypots de production et les Honeypots de recherche. La première vise à réduire la vulnérabilité aux attaques de pirates d'une organisation, la seconde, à obtenir des informations sur la communauté des pirates [2.17]. Comme les IDS, les Honeypots peuvent être placés à des points stratégiques du réseau. Cette position dépendra des objectifs cherchés.

Conclusion

Nous avons présenté dans ce chapitre les systèmes de détection d'intrusion en général, leurs caractéristiques ainsi que leurs classifications. Nous avons vu qu'ils peuvent être rangés selon leurs sources données, leurs méthodes de détection (comportementale, scénario), ainsi que de leurs comportements après une détection (actif ou passif). Ils peuvent également être classifiés selon la manière même d'analyser les données (centralisée ou distribuée).

Nous avons aussi déterminé que selon nos objectifs, nous pouvons déployer les IDS dans trois positions en général (avant ou après le pare-feu, ou sur la DMZ).

Nous avons aussi remarqué que les différents IDS que nous avons étudiés ne détectent pas toutes les attaques. Dans la troisième partie nous essaierons de mettre en place une plateforme de test et présenterons un algorithme pour la détection des attaques ARP dans Snort.

Troisième partie

Mise en place de notre plateforme de test et présentation de notre algorithme

Troisième partie : Mise en place de notre plateforme de test et présentation de notre solution

III.1 Etude du protocole ARP et algorithme proposé

III.1.1 Introduction

Le protocole ARP (Address Resolution Protocol), est un protocole utilisé par IP pour retrouver l'identificateur unique d'un matériel sur un réseau ethernet. ARP offre un mécanisme souple de correspondance entre adresses IP et adresses physiques (appelées adresses MAC : Media Access Control) sur un réseau local (LAN). L'implémentation de l'ARP sur un matériel réseau, ou un système d'exploitation actuel, implique l'utilisation d'un cache ARP qui est une base de données, liant une adresse IP sur le réseau à une adresse MAC, l'identificateur unique d'un matériel réseau [RFC 826]. Ce protocole joue un rôle central dans la transmission réseau.

III.1.2 Le principe du protocole ARP

Quand un ordinateur A dont l'adresse IP est 192.168.1.1 souhaite communiquer avec un autre ordinateur B dont il connaît l'adresse IP (192.168.1.2), en utilisant le protocole IP, il regarde tout d'abord si le destinataire est répertorié dans son cache ARP. S'il l'est, une trame Ethernet est forgée avec l'adresse MAC du destinataire, puis expédiée sur le réseau : les deux équipements peuvent alors communiquer. Sinon, une requête ARP de type broadcast (requête envoyée à toutes les machines du réseau) demandant, en somme, « qui possède l'adresse IP 192.168.1.2 ? ». Le propriétaire de l'adresse IP en question répond alors, par exemple, « c'est moi, et mon adresse MAC est par exemple 00:00:A6:FC:80:37 ». Maintenant la station source est alors en mesure d'envoyer le paquet IP vers la station destinataire.

Nous voyons donc que le principe d'ARP est très simple et très efficace a priori. Mais il y a un problème, aucun mécanisme n'est défini pour sécuriser le protocole ARP. Ainsi si un utilisateur peut forger ses propres paquets ARP dans le réseau local un problème de sécurité va se poser. En effet, si un ordinateur envoie volontairement et en permanence une réponse basée sur une adresse différente de l'adresse de la véritable machine, il pourra détourner les informations vers un autre ordinateur. En particulier, un pirate peut très bien envoyer une réponse ARP à une station en lui disant que, dorénavant, l'adresse IP 192.168.1.2 correspond à sa propre adresse MAC ou à toute

autre adresse MAC de son choix. On dit alors que la station a été « empoisonnée ». L'empoisonnement est rendu d'autant plus facile que la plupart des stations acceptent des réponses ARP, alors même qu'elles n'ont envoyé aucune requête ARP.

III.1.3 Les attaques possibles avec le protocole ARP

Plusieurs types d'attaques peuvent porter sur le protocole ARP. Parmi lesquelles :

- l'usurpation d'adresse MAC (*MAC spoofing*) : son principe repose sur l'envoi d'une trame ayant pour adresse source l'adresse MAC de notre victime, et pour destination notre adresse MAC, le commutateur, en recevant cette trame, met sa table à jour en associant l'adresse MAC de la victime à notre port;
- l'usurpation d'identité ARP (*ARP spoofing*) : que nous allons détailler dans la suite du document;
- la corruption de cache ARP (*ARP cache poisoning*) : son principe est d'agir directement sur la cache ARP de notre cible, en associant à une adresse IP à une adresse MAC inexistante sur le réseau, le pirate interdit à la station empoisonnée tout trafic en direction de l'adresse IP choisie. C'est une attaque de type déni de service (DoS);
- la perturbation de la commutation des commutateurs : le pirate peut bombarder un commutateur (*switch*) avec des milliers de réponses ARP dans le but de saturer sa table ARP. Dans ce cas, certains commutateurs réagissent par la suite en retransmettant tous les paquets qu'ils reçoivent sur tous leurs ports. En d'autres termes, ils se comportent alors en simples répéteurs, leur fonction d'isolation des communications est éliminée.

I.3.1 Mécanisme de l'attaque ARP Spoofing.

Supposons que deux stations T1 et T2 souhaitent communiquer; avant l'attaque nous avons la situation présentée à la figure 3. 1.

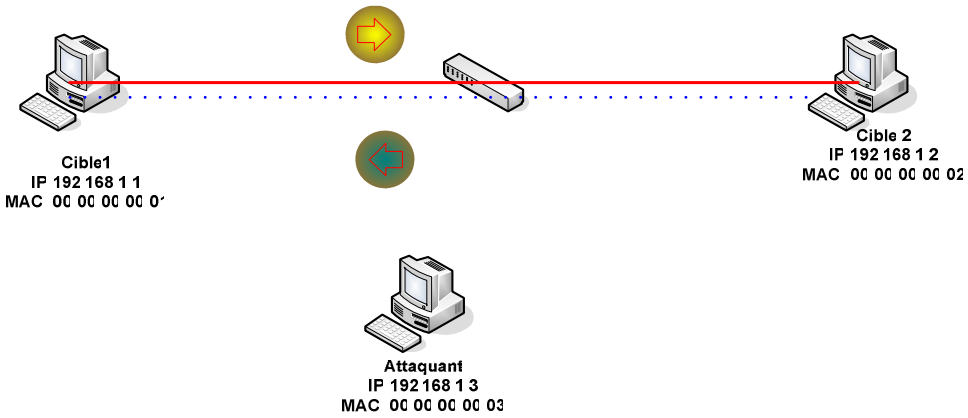


Figure 3. 1: Initiation de ARP Spoofing

Avant l'attaque, T1 et T2 ne communiquaient qu'entre elles. La commande arp –a (sous Linux, Windows, BSD …) permet d'afficher les tables respectives de T1 et T2.

Tableau 6: Table ARP de l'émetteur T1 (adresse IP : 192.168.1.1) avant l'attaque

ADRESSE IP	ADRESSE MAC
192.168.1.2	00 :00 :00 :00 :00 :02
192.168.1.200	00 :00 :00 :00 :00 :04

Tableau 7: Table ARP du récepteur T2 (adresse IP : 192.168.1.2) avant l'attaque

ADRESSE IP	ADRESSE MAC
192.168.1.1	00 :00 :00 :00 :00 :01
192.168.1.200	00 :00 :00 :00 :00 :04

Le commutateur ne manipule que les adresses MAC pour transmettre les paquets aux machines destinataires. C'est pourquoi si l'on parvient à modifier les tables ARP, on peut détourner le trafic. Après l'attaque nous avons la situation illustrée à la figure 3.2.

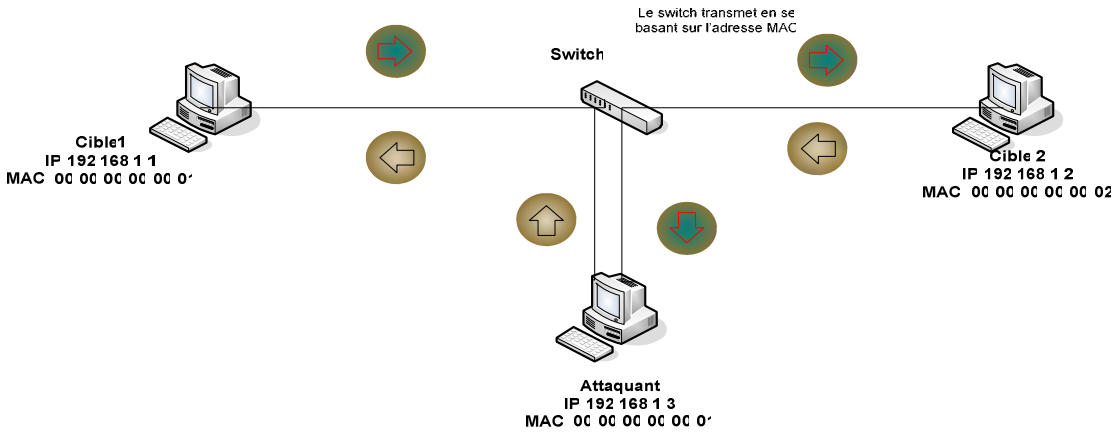


Figure 3. 2: Attaque ARP Spoofing

Les tables ARP des machines T1 et T2 sont présentées dans les tableaux 5 et 6 respectivement.

Tableau 8: Table ARP de l'émetteur T1 (192.168.1.1) après l'attaque

ADRESSE IP	ADRESSE MAC
192.168.1.2	00 :00 :00 :00 :00 :03
192.168.1.200	00 :00 :00 :00 :00 :04

Tableau 9: Table ARP du récepteur de T2 (192.168.1.2) après l'attaque

ADRESSE IP	ADRESSE MAC
192.168.1.1	00 :00 :00 :00 :00 :03
192.168.1.200	00 :00 :00 :00 :00 :04

Dans les deux tables **00:00:00:00:00:03** est l'adresse MAC de la machine attaquante.

Ce type d'attaque peut donc entraîner une indisponibilité du réseau, si toutefois le pirate donne de fausses adresses MAC qui n'existent même pas.

III.1.4 Quelques outils pour les attaques relatives au ARP

Il existe plusieurs outils pour initier des attaques exploitant les insuffisances du protocole ARP. On distingue :

- Ettercap, qui est un outil utile pour les attaques MiM. Il possède aussi des capacités de sniffing, mais utiliser Ethereal pour le sniffing est plus intéressant car il utilise le format

pcap pour enregistrer les paquets. Pcap est un format assez ancien et il existe beaucoup d'outils pour analyser les fichiers pcap ;

- Arp-sk est un outil permettant de forger des trames MAC ;
- Scapy est un forgeur de paquets, qui permet de générer et de recevoir rapidement et avec précision des trames et des paquets de toute une série de protocoles (802.3, 802.1q, 802.11, SNAP, MGCP, RIP, TCP, UDP, BOOTP, DHCP, NTP, NBNS, Radius, SMB, SIP, etc). Il permet aussi de réaliser des tâches classiques de scanning, de tracerouting, de sondage, d'attaque et de découverte de nouveaux réseaux ;
- Arpspoof est un outil qui peut envoyer des paquets ARP reply falsifiés à toutes les machines sur le réseau en les renseignant que l'adresse MAC de la passerelle.

III.2 Mise en place d'une plate-forme de test

III.2.1 Introduction

Pour bien comprendre les mécanismes des attaques et les moyens de défense des IDS il convient de disposer d'un environnement de test approprié. A cet effet nous nous proposons de mettre en place une plate-forme de test avec l'IDS Snort. Des attaques sont simulées à l'intérieur comme à l'extérieur de notre réseau pour voir la fiabilité des alertes remontées par notre IDS. Pour ce faire nous utilisons Nmap, un utilitaire open source très puissant et très connu dans le domaine de la sécurité, développé par Fyodor et est distribué par Insecure. Il permet ainsi, entres autres, de scanner un réseau pour la récolte d'informations (les ports ouverts, les systèmes d'exploitations installés sur les machines cibles).

Rappelons qu'un balayage de ports effectué sur un système tiers est considéré comme une tentative d'intrusion et est donc considéré comme **illégal**. Les mêmes commandes seront exécutées sur la machine du pirate (extérieur à notre réseau) et sur une machine de notre réseau.

III.2.2. Architecture et mise en place de la plate-forme

La plate-forme est composée d'un réseau interne (1921.168.5.0/24), sur lequel est installé sonde (SNORT), d'une autre machine qui appartenant à un autre sous réseau, et d'une passerelle équipée de deux cartes réseau. Une machine avec le système d'exploitation Debian fait office de passerelle et de Firewall. La structure de la plateforme de test est présentée à la figure 3.3

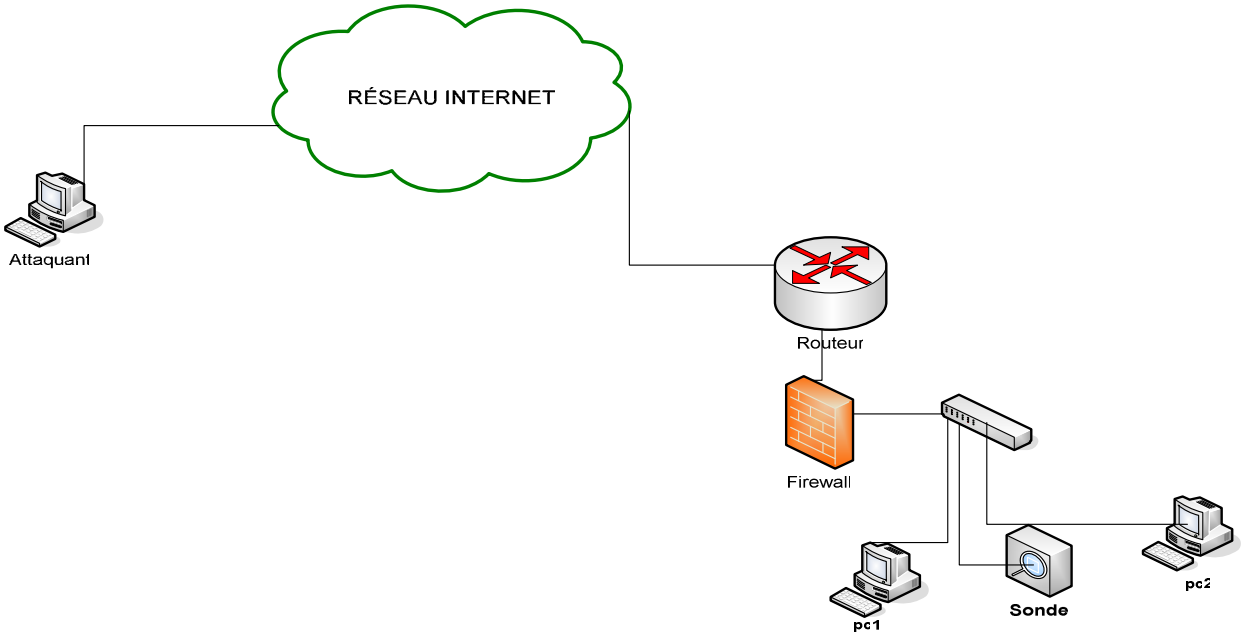


Figure 3. 3: Architecture de la plateforme de test

Pour mettre en place notre passerelle nous avons choisi un ordinateur PIV 2.5GhZ, avec une Ram de 512 Mo et un disque dur de 30 Go, sur lequel nous avons installé Débian (noyau version 2.6.18). Après une mise à jour complète du système, nous avons activé le forwarding sur la machine en modifiant le fichier */etc/sysctl.conf* : *net.ipv4.ip_forward =1*. Après l'activation de forwarding notre machine se comporte exactement comme un routeur.

La mise en place de la passerelle et la définition de la politique de sécurité se sont faites avec iptables. Le script pour désactiver le Firewall est enregistré dans un fichier nommé **firewall-stop,** et est placé dans le répertoire **/etc/firewall-stop.**

```
#!/bin/sh

# Nous vidons les chaînes :
iptables -F
# Nous supprimons d'éventuelles chaînes personnelles :
iptables -X

# Nous les faisons pointer par défaut sur ACCEPT
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

# Cette ligne permet de supprimer toutes les redirections de ports
iptables -t nat -F
```

Le fichier qui démarre le Firewall est nommé **firewall-start** et placé dans le répertoire **/etc/firewall-start.** Ce fichier est présenté dans l'annexe E.

III.3 Installation et configuration des composants logiciels

Cette partie est constituée de plusieurs étapes :

- Installation des paquetages pré-requis ;
- Installation et configuration de la sonde (SNORT) ;
- Installation et configuration de l'interface BASE.

III.3.1 Installation et configuration des paquetages

Avant de commencer l'installation de SNORT, les packages suivants doivent être installés :

- **fedora core 6** : aussitôt après l'installation nous avons fait une mise à jour complète de notre système ;
- **MySQL** : la base de données MySQL, (SQL Structured Query Language en anglais) que nous utilisons pour stocker les logs et les alertes remontées par notre sonde. En plus d'être très rapide, robuste et multi-utilisateurs. MySQL est un logiciel libre développé sous licence GNU (General Public License). On doit aussi installer le module **mysql-devel ;**
- **mysql-client** : la partie cliente de MySQL (connexion à la base de données) ;
- **php-mysql** : qui est le module PHP de MySQL ;
- **Apache** : un serveur web de même que le module de PHP pour Apache. En effet Apache est un logiciel qui sert des requêtes respectant le protocole de communication Client-Serveur : HyperText Transfer Protocol (HTTP), développé pour le World Wide Web ;
- **PCRE** (Acronyme de Perl Compatible Regular Expression, soit une expression régulière compatible avec le Perl). Nous avons téléchargé et installé **pcre-7.0.tar.gz** ;
- **php-pear :** les composants de PEAR (PHP Extension and Application Repository) constituent une bibliothèque de codes, qui permet de réduire le temps de développement par le biais de classes préécrites pour les fonctionnalités standard. Il assure le bon fonctionnement de BASE, que nous présentons dans la suite du document;
- **php-gd :** PHP avec la bibliothèque **GD ;**
- **Libpcap / libpcap−devel :** ce sont des bibliothèques de fonctions très puissantes qui servent d'interface pour la capture de paquets et sont indépendantes du système ;
- **Gcc** : est un compilateur libre pour différents langages comme `C` ou `C++` mais aussi pour JAVA, indispensable pour compiler les sources de SNORT.

III.3.2 Installation de SNORT et Configuration de SNORT

Installation : Après avoir téléchargé la dernière version de Snort à l'adresse http://www.snort.org/dl/, nous avons procédé à son installation. Cette installation est présentée d'une manière très détaillée dans l'annexe A.

Configuration de Snort : Tout d'abord nous avons édité le fichier de configuration de SNORT *(/etc/snort/snort.conf),* et nous sommes passé aux étapes suivantes :

- ✓ Spécification du réseau que SNORT doit écouter : il faut pour cela modifier la variable var HOME_NET :

  var HOME_NET 192.168.5.0/24 # SNORT écoute le réseau 192.168.5.0

  var HOME_NET [192.168.5.0/24 ,10.0.0.1/24] # SNORT écoute deux réseaux à la fois ;

  Nous pouvons aussi entrer une liste d'adresses entre deux crochets et séparées par des virgules (sans espace) :

  var HOME_NET [192.168.5.2 ,10.0.0.2,172.16.30.25]
- ✓ Modification de la variable var RULE_PATH ../rules par *var RULE_PATH /etc/snort/rules* ;
- ✓ Modification de la variable: *preprocessor stream4_reassemble : both, ports 21 23 25 53 60 110 111 139 143 445:*

  Stream4 fut initialement conçu pour protéger Snort d'un nouveau genre d'agressions qui attaquaient les environnements IDS en les inondant avec des paquets contenant des chaînes de caractères susceptibles de déclencher des alertes. Ces types d'attaques furent relevés avec l'apparition de deux outils appelés *stick et snot*.

  both : Réalise le ré-assemblage pour les deux cotés de la conversation. Si notre sonde Snort surveille des clients et des serveurs, nous avons besoin d'activer cette option.

  Port : spécifie la liste de ports dont le traffic sera ré-assemblé.
- ✓ Liaison des logs de SNORT à MySQL : *output database:log, mysql, user=snort password=snort1passe dbname=snort host=localhost*

host indique l'adresse IP du serveur de base de données, si elle n'est pas précisée, le système local est la valeur par défaut ;

dbname est le nom (quelques fois appelé « instance ») de la base de données sur le serveur de base de données ;

user indique le nom d'utilisateur utilisé par Snort pour se connecter au serveur de base de données ;

password est le mot de passe utilisé pour se connecter au serveur de base de données.

log/alert : Avec log, les journaux de capture de paquets sont envoyés dans la base de données ; avec alert, c'est le flux des alertes qui y est enregistré. Log inclut les informations sur les alertes ainsi que sur les paquets qui déclenchent les alertes. C'est l'option la plus souvent utilisée. Pour envoyer les deux types de données vers la base de données, il faut préciser deux lignes output database, chacune contenant l'une des options.

III.3.4 Création de la base de données de Snort

Ensuite nous avons créé la base de données utilisée comme emplacement de stockage des logs et des alertes de Snort. La consultation de ces derniers se fait via l'interface web de BASE, qui est entièrement développée en PHP. BASE est disponible sur *http://base.secureideas.net/*.

```
$] # service mysql start   // Démarrer MySQL
$] # mysql –u root –p  // connexion au serveur MySQL avec les privilèges de root
> create database snort; //creation de la base de données de stockage des alertes et log
> grant create, insert, select, delete, update on snort.* to snort ;
> grant create, insert, select, delete, update on snort.* to snort@localhost ;
> set password for snort@localhost=PASSWORD ('snort1passe');
> set password for snort@%=PASSWORD ('snort1passe');
> flush privileges ;
> exit ;
```

La base de données snort est créée mais sans les tables qui en constituent la structure.
Sur certains systèmes, l'erreur suivante peut survenir :
Can't connect to local MySQL server through socket '/tmp/mysql.sock'.

Cela arrive lorsque le client MySQL cherche le socket du serveur MySQL dans le répertoire */tmp* alors qu'il se trouve dans */var/lib/mysql*. Pour résoudre ce problème, nous pouvons utiliser les commandes suivantes:

$] # cd /tmp

$] # ln –s /var/lib/mysql/mysql.sock mysql.sock

Vérifions avec la commande **ls –la** que le fichier */tmp/mysql.sock* est un lien symbolique pointant vers son homonyme du répertoire de MySQL.

Pour créer les tables de la base de données avec un schéma déjà existant nous utilisons la commande suivante:

$] # mysql –u root –p < /usr/local/src//snort-2.6.1.4/schema/create_mysql snort

La liste des tables créées est présentée dans l'annexe C.

Nous voulons assurer le démarrage de Snort en automatique. Tout d'abord nous devons récupérer le script nécessaire pour faire fonctionner Snort sous forme de service à partir de http://www.ewatching.net/projet/snort.txt. Après avoir récupéré le script nous le copions dans le répertoire */etc/init.d/snort* ; ensuite nous exécutons les commandes suivantes :

$#] cd /etc/init.d

$#] chmod 755 snort

Maintenant deux possibilités s'offrent à nous : Soit nous lançons Snort tout seul, et dans ce cas, il générera ses logs dans un fichier ; soit nous décidons de l'interfacer avec une base de données. Suivant le cas, Snort ne se lancera pas de la même façon.

Sans Mysql : *#] /usr/local/snort*/src/snort −c /etc/snort/snort.conf −i eth0 −D*

Avec Mysql : *#] /usr/local/snort*/src/snort −c /etc/snort/snort.conf*

III.3.5 Installation de BASE et d'ADODB

Ces composants permettent une bonne exploitation des logs créés par SNORT.

III.3.5.1 Installation de BASE

BASE est une interface PHP qui permet de visualiser les remontées d'alarmes générées par SNORT. On peut le télécharger à partir de http://downloads.sourceforge.net/secureideas/.

```
$] # mv base-1.3.8.tar.gz /var/www/html
$] # tar –zxvf base-1.3.8.tar.gz
$] # rm –f  base-1.3.8.tar.gz
$] # mv base-1.3.8 base
$] # mv base_conf.php.dist  base_conf.php
$] # vi snort/base_conf.php
```

Il faut aussi installer quelques composants graphiques de Base pour son bon fonctionnement grâce aux commandes suivantes :

```
$] # pear install Image_Graph
$] # pear install Image_Canvas
$] # pear install Image_Color
$] # pear install Numbers_Roman
```

III.3.5.2 Configuration de Base

La configuration de Base consiste à éditer le fichier base_conf.php se trouvant dans */var/www/html/base* et à modifier certaines lignes. Les détails sont présentés dans l'Annexe D.

Ensuite il faut télécharger et installer ADODB. En effet ADODB contient des scripts PHP génériques de gestion de bases de données. Son installation est simple, car il suffit de le copier dans le répertoire racine d'apache (*/var/www/html/adodb*). On peut le télécharger à partir de http://easynews.dl.sourceforge.net/sourceforge/adodb/adodb462.tgz.

```
$] # mv adodb462.tgz /var/www/html
$] # cd /var/www/html
$] # tar –zxvf adodb462.tgz
$] # rm –f adodb462.tgz
```

Pour visualiser les logs créés par SNORT, il nous faut juste ouvrir un navigateur web (Mozilla, firefox, Internet explorer…), et taper : http://localhost/base. La page d'accueil de BASE est présentée dans l'annexe B

III.4. Tests sur les scans de port

Nous allons maintenant lancer une série de tests à destination de Snort pour voir son comportement par rapport à certaines attaques. Les tests sont réalisés avec TCP *connect() scan* et *protocol scan.*

Les paquets envoyés par le pirate pour récolter des informations sur la machine cible sont normalement journalisés dans des fichiers logs qui se trouve dans le répertoire */var/log/snort.*
Ces fichiers sont ensuite analysés par l'administrateur afin de prendre les mesures qui s'imposent. Cette procédure est ennuyeuse et demande beaucoup de temps pour faire ressortir des informations utiles, surtout quand la taille des fichiers log explose.

III.4.1 TCP connect () scan.

Au niveau de la machine pirate la commande ***nmap –ST 192.168.5.0/24,*** donne les résultats de figure 3.4.

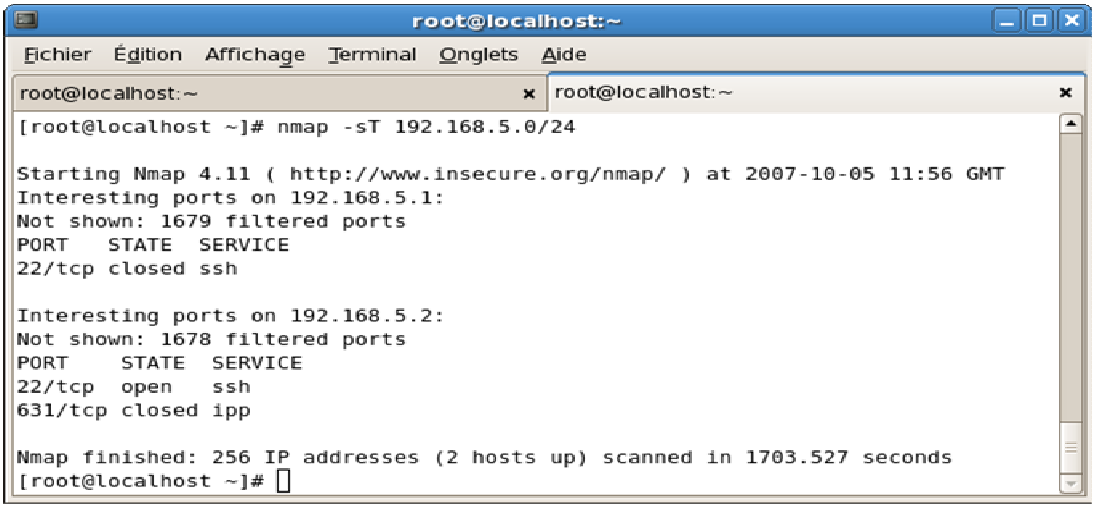


Figure 3. 4: Scan de type TCP connect 1

Nous constatons que le pirate n'a pas obtenu suffisamment d'informations sur les machines du réseau 192.168.5.0/24, alors que la figure 3.5 obtenue par capture au niveau de la machine locale montre des informations plus intéressantes.

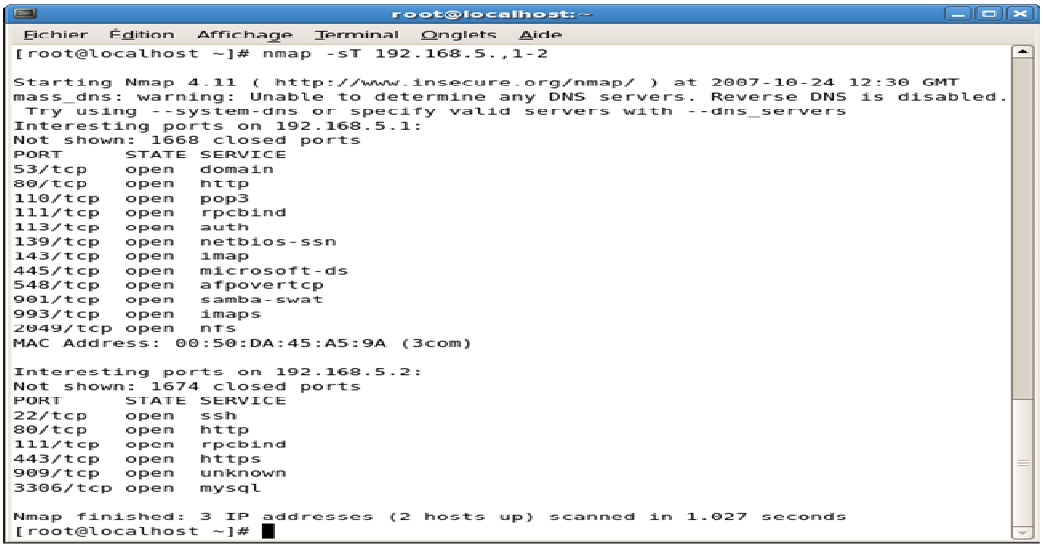


Figure 3. 5: Scan de type TCP connect 2

Nous constatons aussi des alertes au niveau de notre sonde à la figure 3.8

III.4.2 Test protocol scan

Au niveau de la machine pirate la commande ***nmap –sO 192.168.5.2,*** donne les résultats que nous avons à la figure 3.6.

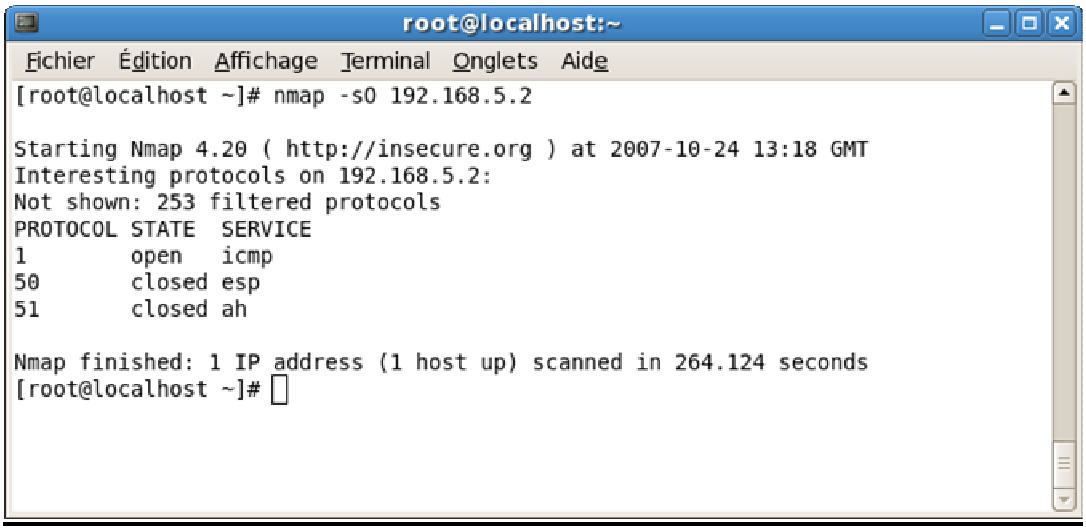


Figure 3. 6: Test protocole scan 1

Ici la figure 3.6 donne seulement des informations sur 3 ports. En local la même commande nous donne la figure 3.7, ce qui permet d'obtenir beaucoup plus d'informations.

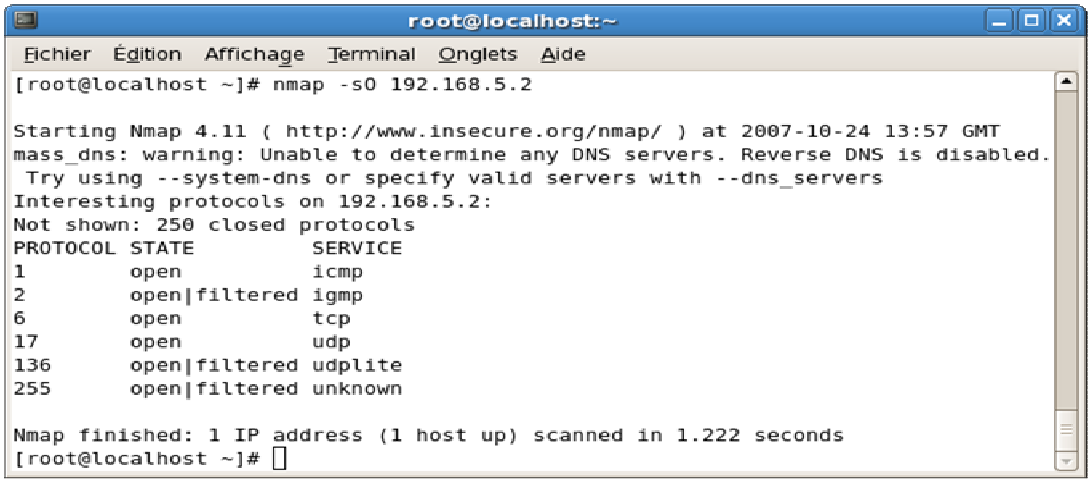


Figure 3. 7: Test protocole scan 2

Nous remarquons que Nmap peut donner des informations sur l'ouverture (open), la fermeture (closed) ou sur le filtrage (filtered) d'un port.

III.4.3 Détail d'une alerte au niveau de l'interface web de base

La capture (figure 3.8) suivante nous donne des détails sur une alerte.

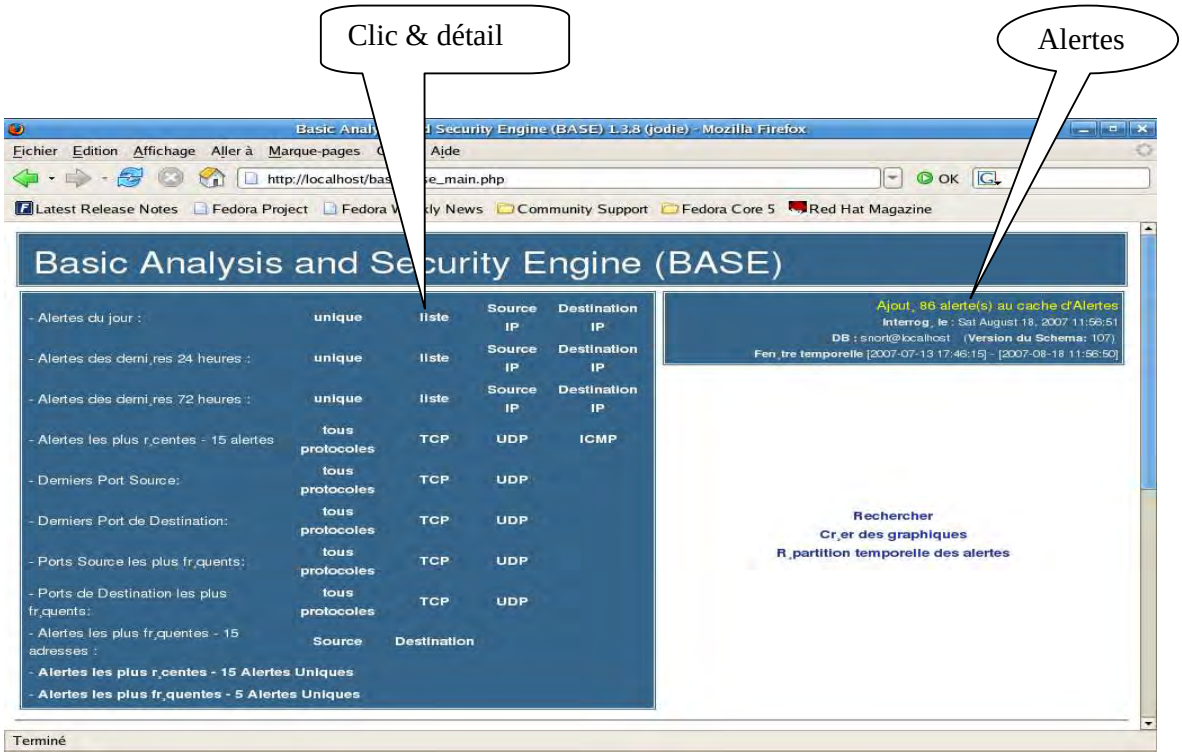


Figure 3. 8: Interface web de base avec des alertes remontées

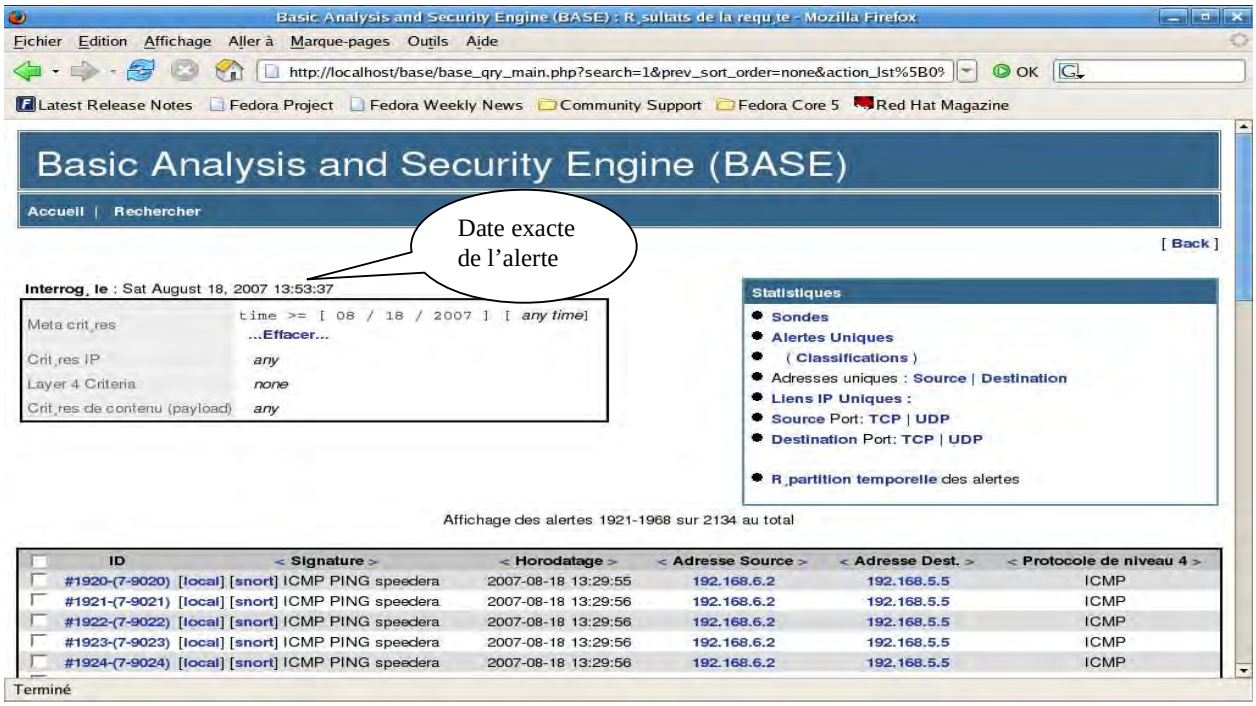


Figure 3. 9: Informations détaillées sur les alertes

Cette figure 3.9 nous donne des informations sur plusieurs alertes, comme :

- le jour est l'heure exacte de l'attaque ;
- la signature qui a déclanché l'alerte ;
- l'adresse source de l'attaquant ;
- l'adresse de la cible ;
- le type de protocole utilisé pour réaliser l'attaque.

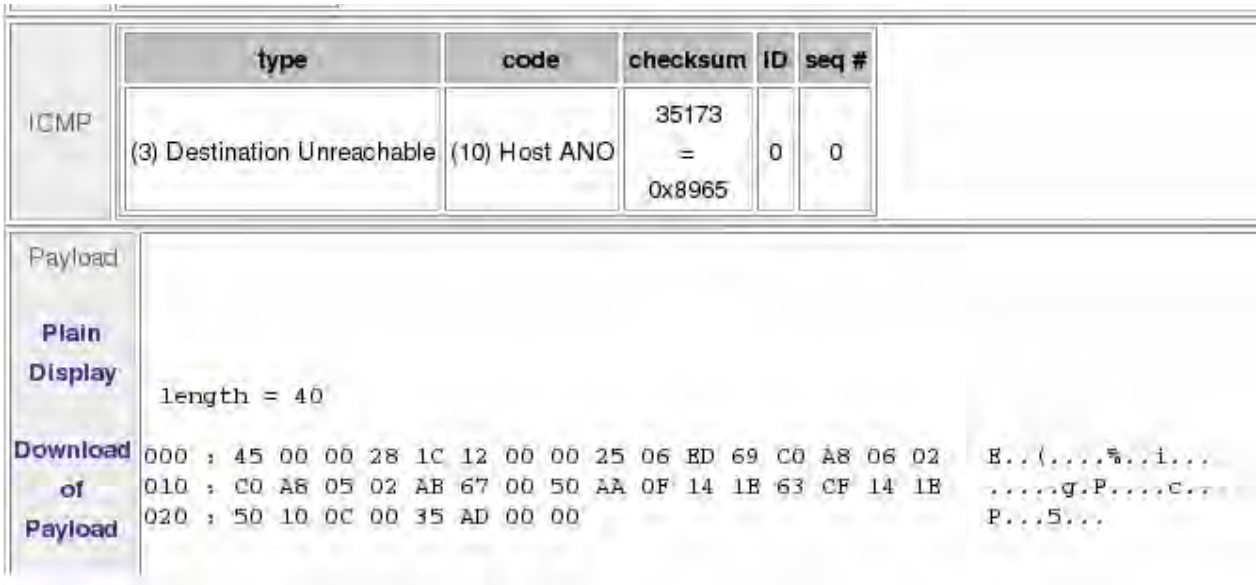


Figure 3. 10 : Informations détaillées sur une alerte particulière

Cette figure 3.10 donne des informations sur le contenu d'un paquet suspect.

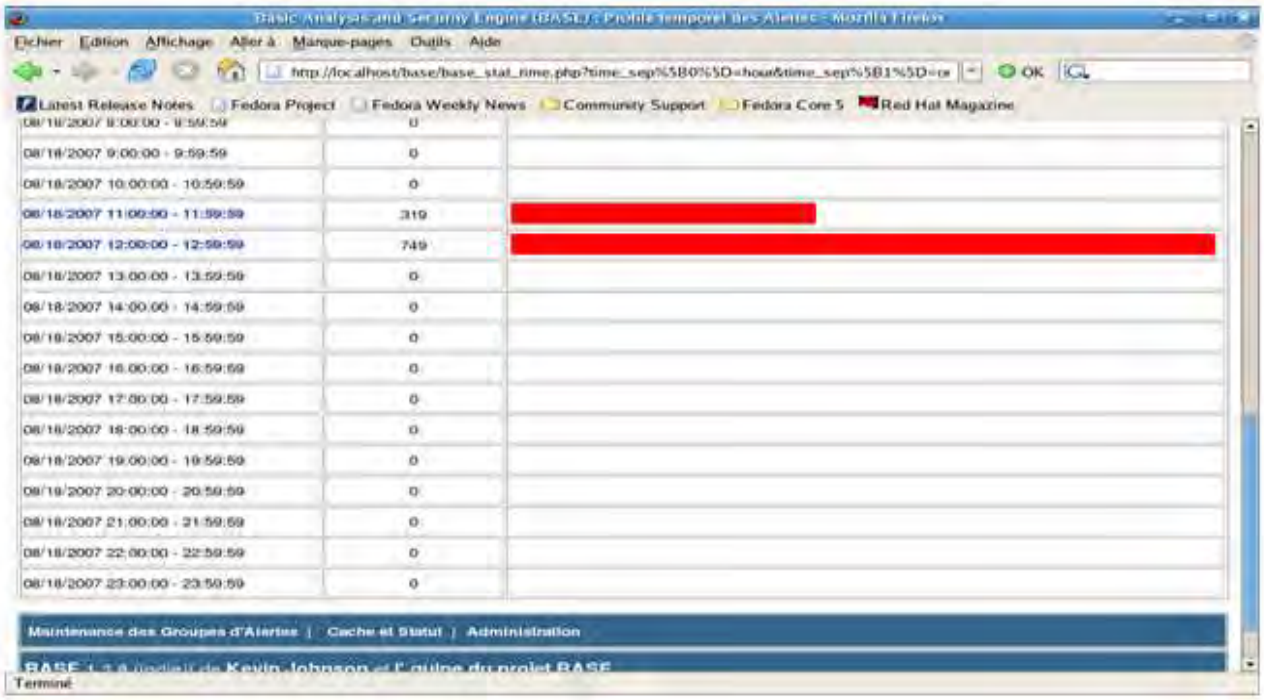


Figure 3. 11: Alertes remontées par date (jours, heures)

La figure 3.11 nous donne des informations sur le nombre d'alertes remontées en une heure, elle nous donne en même temps une représentation graphique des attaques.
Nous pouvons aussi obtenir des informations plus détaillées sur les alertes remontées par la sonde.

La capture suivante (figure 3.12) est une interface qui permet une représentation graphique des alertes en fonction des options cochées. A partir de cette figue nous pouvons avoir comme information :
Après avoir choisi les paramètres que nous voulons observer, il nous faut juste cliquer sur le bouton **Graph Alerts** pour voir les résultats.

Nous signalons que la réalisation de cette interface pour la génération des graphes (figure 3.13, 3.14, 3.15), nous a causé beaucoup de problèmes, mais que nous avons surmonté.

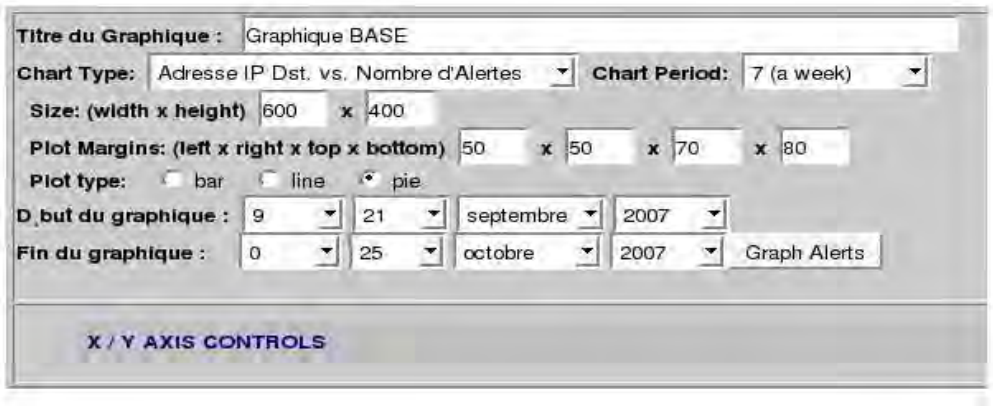


Figure 3. 12: Choix du mode d'affichage des alertes

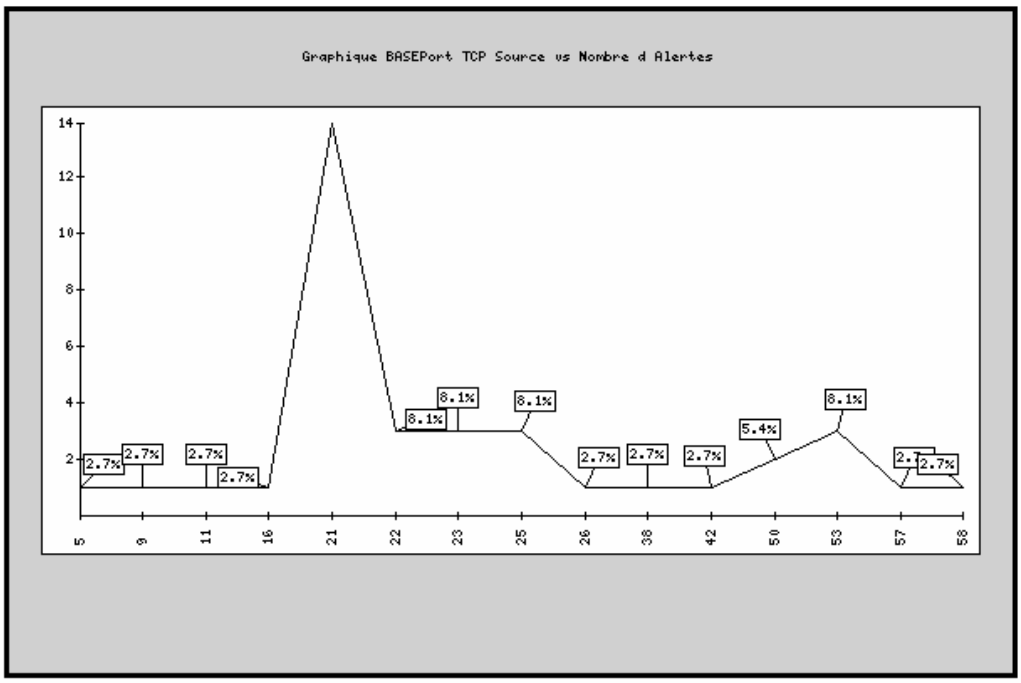


Figure 3. 13: Nombre d'alertes remontées TCP en fonction du temps

Cette figure 3.13 donne une représentation graphique des attaques initiées par une machine source en utilisant le protocole TCP, et en fonction du nombre totale d'alertes remontées entre les dates choisies dans la figure 3.12.

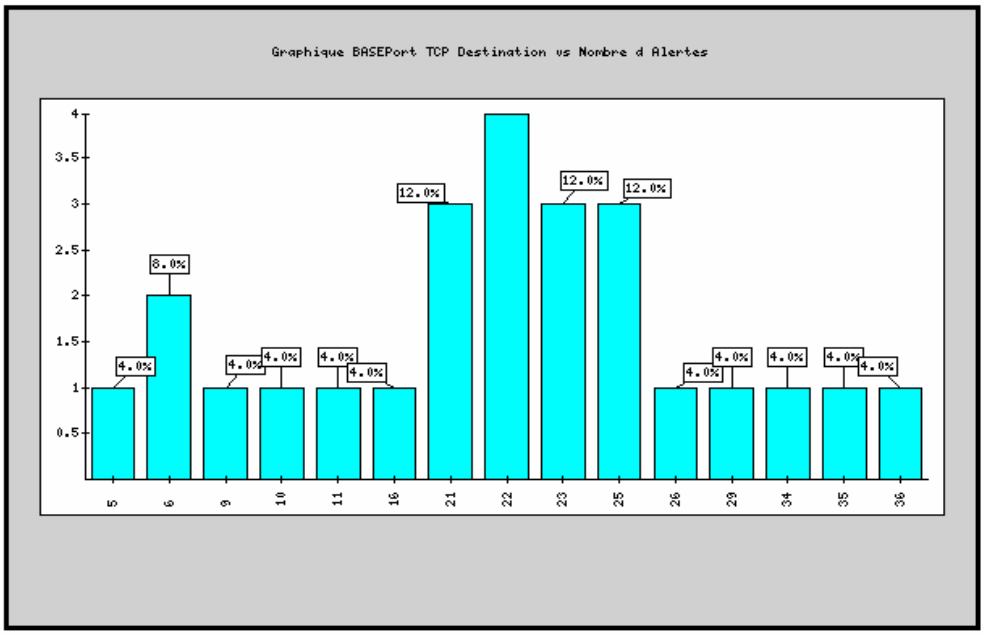


Figure 3. 14: Nombre d'alertes remontées en fonction du temps

Cette figure 3.14 donne une représentation graphique des paquets envoyés par la cible, en utilisant le protocole TCP, et en fonction du nombre totale d'alertes remontées entre un intervalle de temps spécifique.

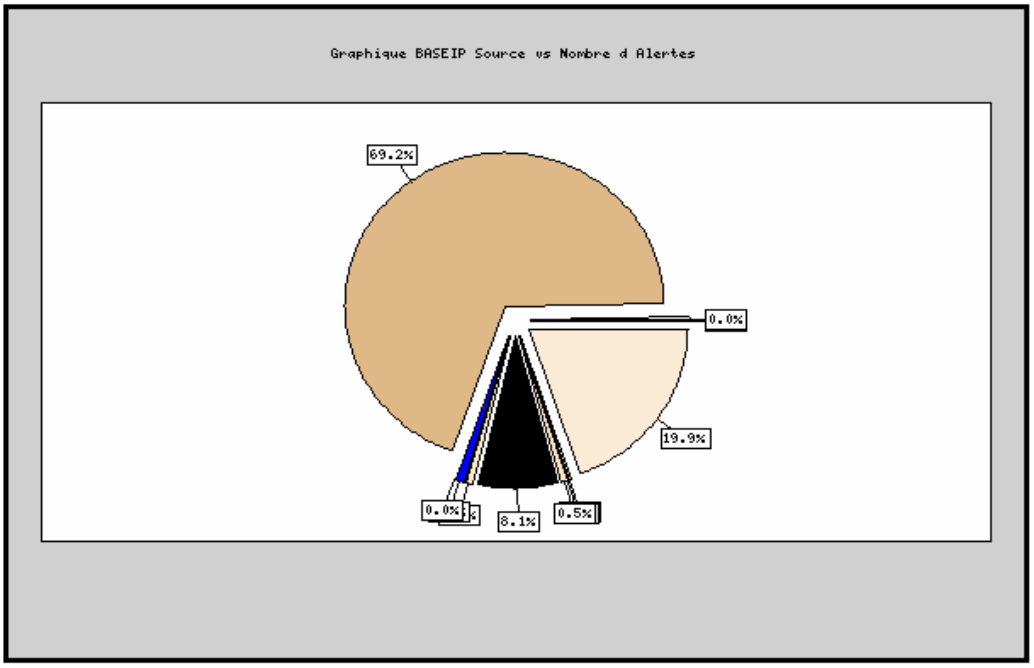


Figure 3. 15: Nombre d'alertes remontées par une source en fonction du temps

Cette figure 3.15 donne une représentation graphique des paquets IP envoyés par une source en fonction du nombre totale d'alertes remontées entre un intervalle de temps spécifique.

Nous pouvons aussi avoir d'autres informations relatives aux nombres de paquets reçus, traités supprimés, ou concernant les protocoles utilisés dans tout le réseau (figure 3.16).

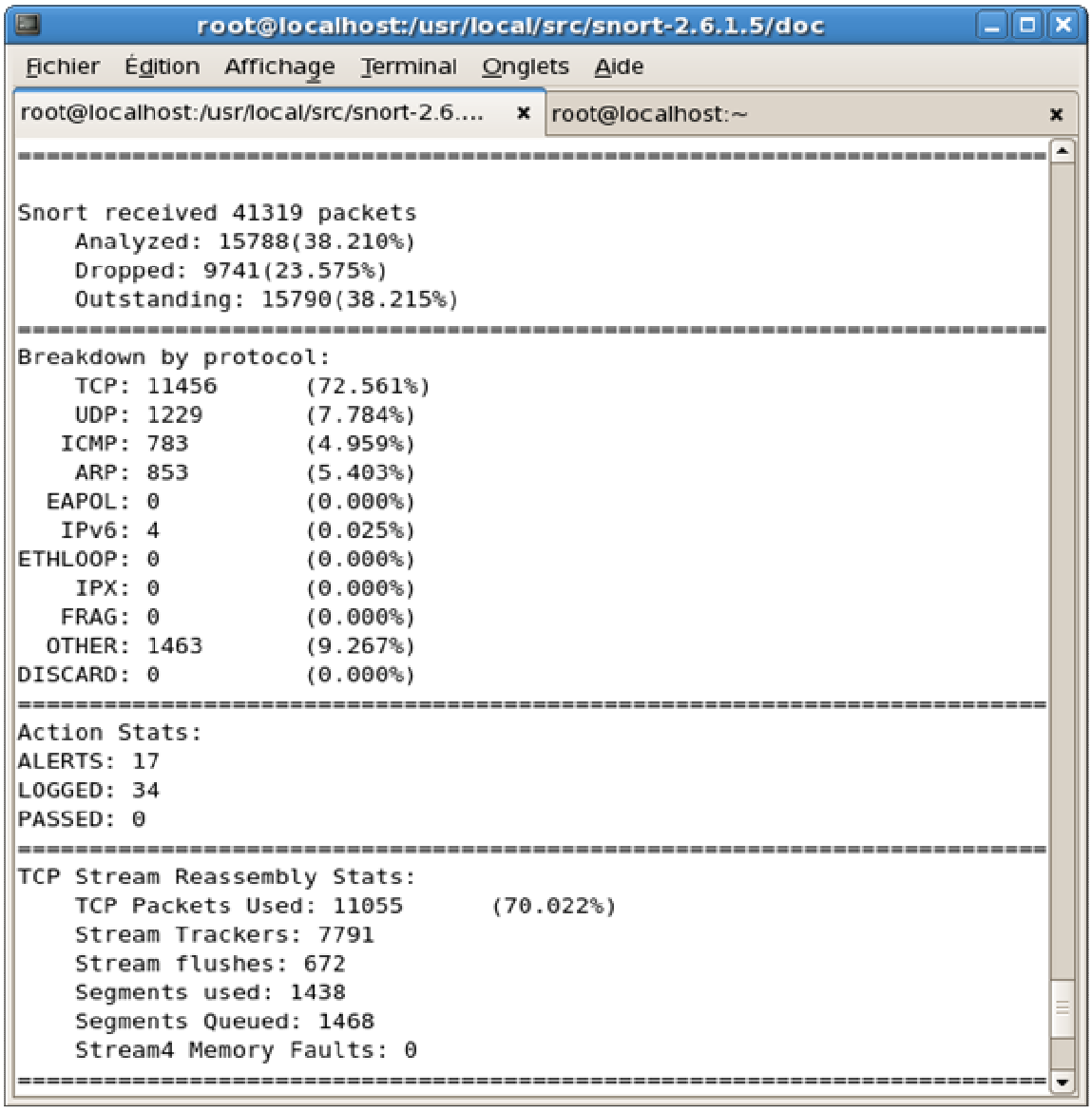


Figure 3. 16 : Statistiques sur le trafic réseau après arrêt de Snort

Ces statistiques nous montrent que sur 41319 paquets reçus par la sonde il n'y a que 15788 (38%) paquets qui ont été analysés, et 24% des paquets sont supprimés. Ceci va sans doute entraîner beaucoup de faux positifs et faux négatifs.

Conclusion partielle

Pour réaliser ce travail, de nombreux problèmes ont été rencontrés et surmontés. Principalement, des difficultés rencontrées lors de l'installation des différents rpm et le nombre important des packages complémentaires à Snort qu'il faut installer et bien configurer.
Nous avons testé notre plateforme en simulant des attaques sur notre réseau interne et nous avons évalué le comportement de notre sonde.
C'est ainsi que nous avons constaté que la plupart des attaques que nous avons simulées ont été détectées par notre IDS.
Mais le nombre très élevé d'alertes remontées relatives à une seule attaque est un grand problème, et pour un trafic réseau soutenu, notre sonde n'arrive pas à analyser l'ensemble des paquets.

III.5 Architecture et fonctionnement de Snort

Avant d'essayer de doter Snort de mécanismes lui permettant de résister aux diverses attaques il est primordial de comprendre sa structure et son fonctionnement. La figure 3.17 nous montre la structure de Snort.

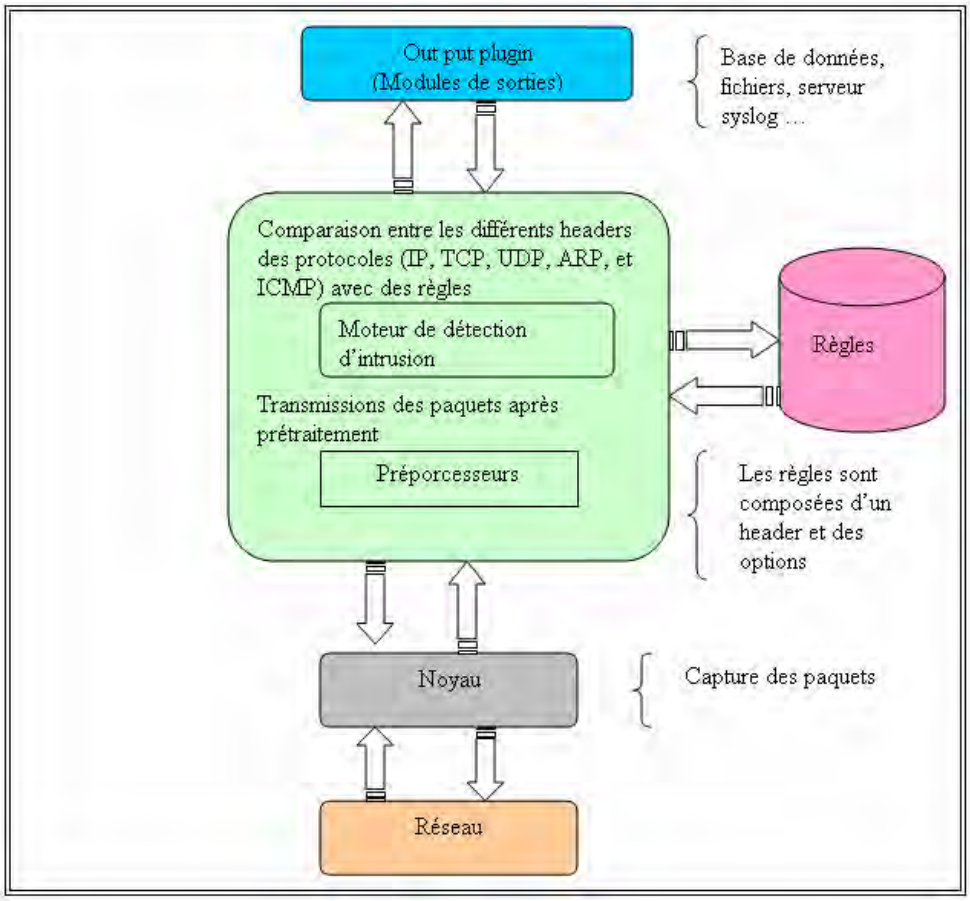


Figure 3. 17: Mécanisme de fonctionnement de Snort

Nous constatons que Snort n'intègre pas de Préprocesseur supportant des règles relatives au protocole ARP comme les autres protocoles (TCP, UDP, ICMP). Il existe néanmoins un module se basant sur la cohérence des messages ARP pour détecter les attaques le concernant. Mais ce module génère beaucoup de fausses alertes. C'est ainsi que nous avons proposés un Préprocesseur se basant sur des règles pour la détection des attaques relatives à ARP.

Pour bien comprendre le rôle du Préprocesseur, il est nécessaire de connaître le mécanisme de fonctionnement de Snort présenté dans la figure 3.17. Il est important de souligner la présence d'un décodeur de paquets dans le noyau. **Un décodeur de paquets,** est composé de plusieurs sous décodeurs qui sont organisés par protocole (Ethernet, IP, TCP..). Ces décodeurs transforment les éléments des protocoles en une structure de données interne.

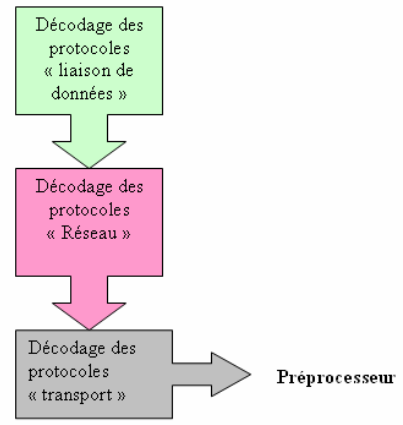


Figure 3. 18 : Décodeur de paquets

Il est important de comprendre qu'un jeu de règles devra être mis en place en fonction du réseau, car toutes les règles fournies ne sont pas forcément utiles au style du réseau, voire même des fois obsolètes. Ainsi les règles sont classées par genre et regroupées dans des fichiers différents. Le dossier rules contient beaucoup de fichiers le montre la figure 3.19 suivante.

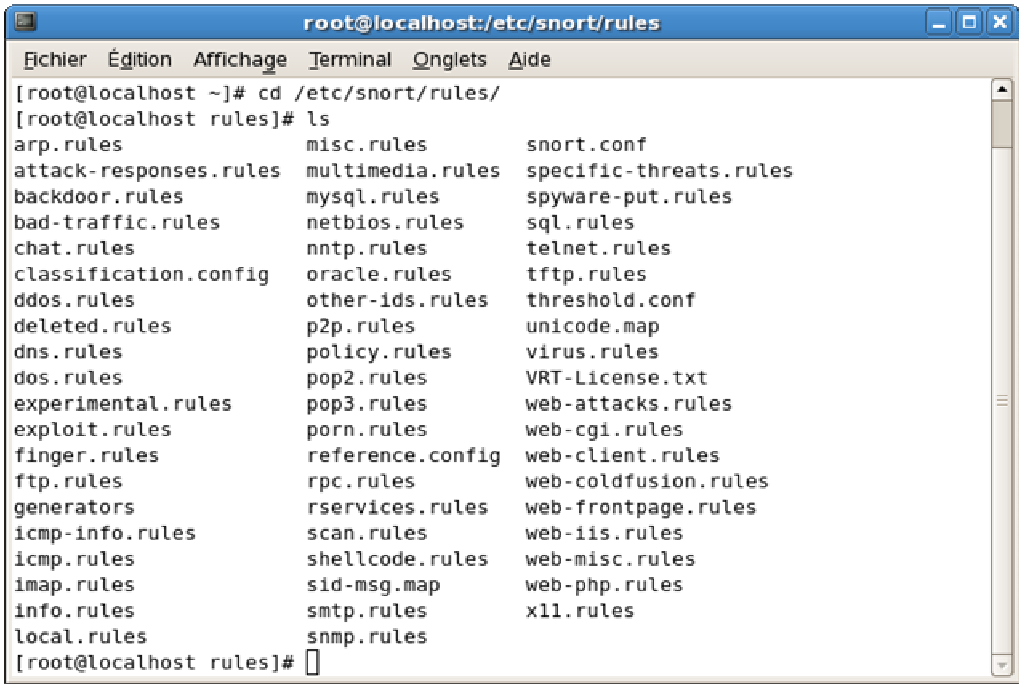


Figure 3. 19 : Les fichiers prédéfinis de Snort du dossier rules

Ces règles ne sont pas activées automatiquement, il faut les activer dans le fichier de configuration *snort.conf*. Chaque fichier contient des règles décrites, et qui signalent un type de trafic. Comme nous pouvons le voir dans le fichier *snort.conf (*figure 3.20).

Notre programme va prendre en paramètre le dossier *rules* qui stocke l'ensemble des signatures d'attaque de Snort. Ces signatures d'attaques vont être comparé avec les paquets capturés dans le réseau pour décider de remonter des alertes ou non.

En effet, Snort bénéficie d'un site officiel sur lequel sont régulièrement mises à jour les nouvelles signatures correspondantes à de nouveaux types d'intrusion. La signature est en général une chaîne de caractères, que nous recherchons à l'intérieur d'un paquet de données. Par exemple la présence de "/scripts/iisadmin/default.htm" dans un paquet à destination d'un serveur Web IIS (Internet Information Services), pourra indiquer une intrusion (tentative d'accès à la page d'administration de IIS).

Figure 3. 20 Une partie du fichier snort.conf

Bien que le site officiel de Snort propose des règles prêtes à l'emploi et régulièrement mises à jour, il peut être intéressant de créer ses propres règles afin d'adapter au mieux Snort au réseau qu'il doit surveiller et protéger.

Une règle Snort est composée de deux parties. Son format est : Header et Options. Nous présentons dans la capture de la figure 3.20 un fichier qui correspond aux règles définies pour le protocole ICMP par exemple.

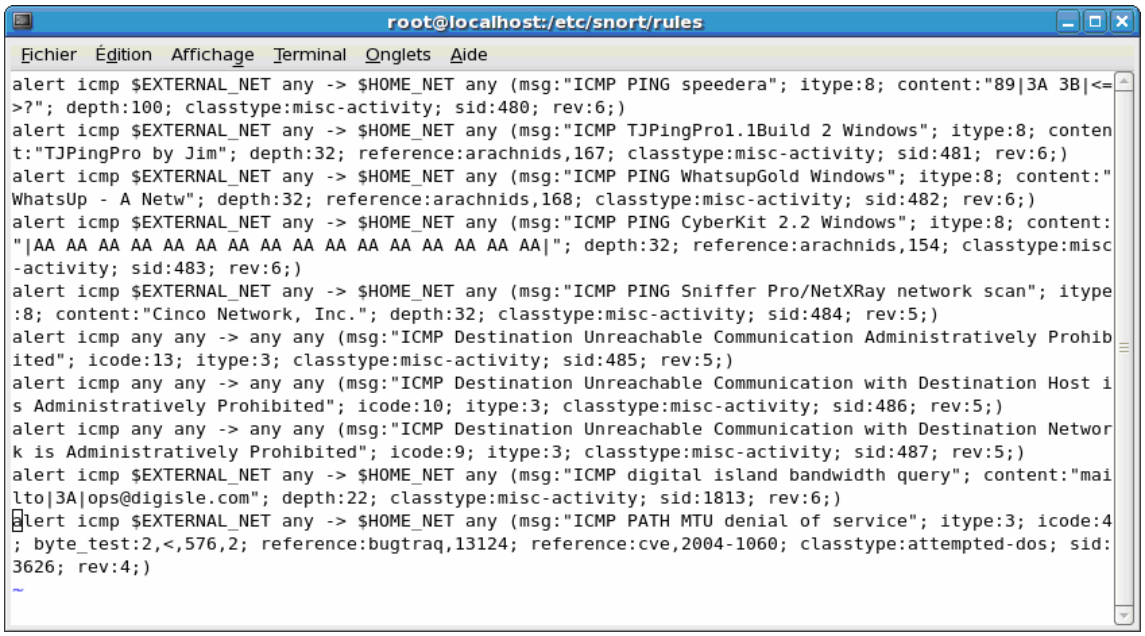


Figure 3. 21 : Structure d'un fichier de règle

Le format de la partie Header est défini de la manière suivante :

action protocole adresse 1 port1 direction adresse2 port2

Le champ **action** peut prendre les valeurs suivantes :

- **alert** : générer une alerte + logger le paquet ;
- **log** : logger le paquet ;
- **pass** : ignorer le paquet ;
- **activate** : activer une règle dynamique ;
- **dynamic** : définir une règle dynamique, qui est passive tant qu'elle n'est pas activée par une autre règle ;
- **drop** : demander à iptables de bloquer le paquet, puis le logger ;
- **reject** : demander à iptables de bloquer le paquet, puis le logger, et envoyer une commande TCP RST (reset) ou une réponse ICMP Host unreachable ;
- **sdrop** : demander à iptables de bloquer le paquet. Ce dernier n'est pas loggé.

Le champ **protocole** spécifie le protocole pour lequel la règle s'applique. Les champs **adresse1** et **adresse2** indiquent les adresses IP source et destination du paquet. Le mot clé **any** permet de spécifier une adresse quelconque. Le champ **direction** spécifie l'orientation du paquet. Cet opérateur peut prendre deux valeurs :

-> : adresse1 vers adresse2

< > : de adresse1 vers adresse2, ou de adresse2 à adresse1

Comme exemple de règle nous pouvons aussi avoir :

Alert tcp any any -> 192.168.5.0/24 80 (flags :A ;\content : “passwd”; msg: “detection de `passwd’ “ ;)

Le texte jusqu'à la première parenthèse est le Header de la règle, et la section entourée par les parenthèses définit les options de la règle. Cette règle permet donc de générer un message d’alerte “détection de passwd” lorsque le trafic à destination d’une machine du réseau local 192.168.5.0/24 vers le port 80, contient la chaîne « passwd » (spécifié par l’utilisation du mot-clé « content »), et que le flag ACK du header TCP est activé (flags : A).

Dans le cadre de notre travail il s’agit d’écrire des règles pour la détection des attaques relatives à ARP. Nous devons ainsi créer le fichier *arp.rules* dans lequel nous spécifions des signatures d’attaques relatives au protocole ARP. Nous pouvons écrire la règle suivante :

alert arp $EXTERNAL_NET any -> $HOME_NET any (msg:"une attaque ARP "; content:"|00 00 00 FF FF 00 00 00 00 ";)

Cette règle permet de générer un message d’alerte “une attaque ARP” lorsque le trafic provient de n’importe quelle adresse source (EXTERNAL_NET), est à destination des machines du réseau local, vers n’importe quel port, et contient la chaîne «00 00 00 FF FF 00 00 00» (spécifié par l’utilisation du mot-clé « content »). L'option de règle *msg* dit au moteur de journalisation et d'alerte quel est le message à imprimer, avec une sauvegarde du paquet ou une alerte. L’option *content* contient la signature qui décrit l’attaque.

Pour écrire des règles pertinentes il faut bien connaître la nature et les particularités du réseau sur lequel nous avons installé Snort. Il existe des utilitaires de supervision réseau comme **Ntop**, **Ethereal** qui peuvent nous aider dans la mise en place de ces règles.

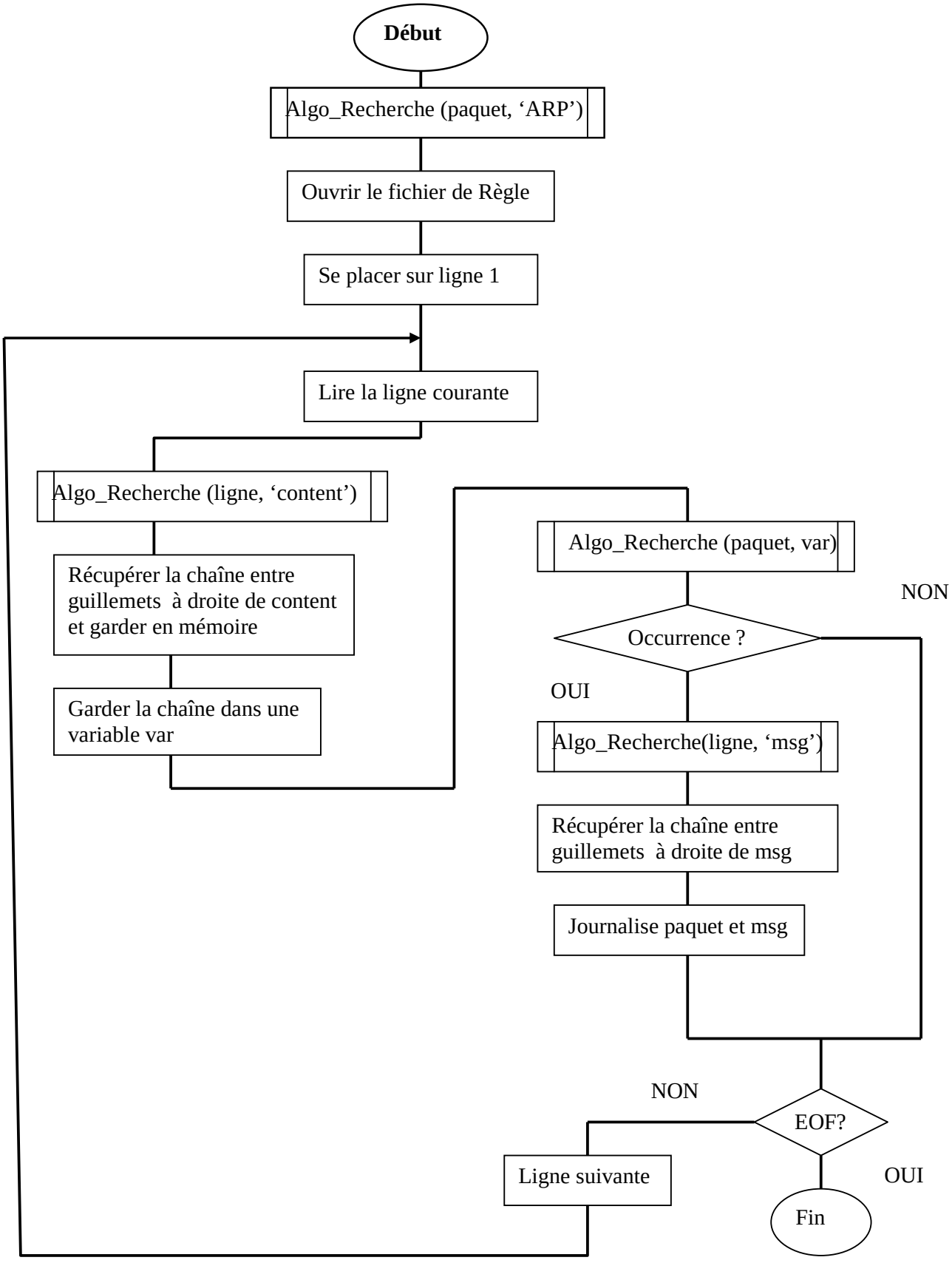
Cette analyse du fonctionnement de Snort montre que le préprocesseur servant à intégrer le protocole ARP dans l’IDS doit rechercher certains motifs (arp, content, etc.) dans le fichier **log** et effectuer des actions concrètes. L’algorithme de recherche de motif dans un texte proposé est le suivant.

Algorithme : Algorithme de recherche de motif dans un texte. Cet algorithme prend en paramètre un texte et un motif.

```
Const        T : texte de longueur n
             P : motif de longueur m
Var          i, j, : entier
Début
    fonc Algo_Recherche (T,P)
        pour j = 0 à n-m
        i := 0
         Tant Que (T [j+i]=P[i] et i<m)
            i = i+1
            Si m = i alors
              garder j en mémoire
            FinSi
          Fin Tant Que
        FinPour
Fin
```

Cette fonction sera par la suite appelée à chaque fois que nous voulons rechercher un motif dans un texte.

Un motif représente une signature ou une anatomie d'attaque alors que un texte représente un paquet capturé dans le réseau.



Algorithme de détection des attaques ARP dans le NIDS Snort

L'algorithme intervient juste après le décodeur de paquet dans la structure interne de Snort. S'il s'agit d'un paquet ARP le programme ouvre le fichier de règle relatif au protocole ARP que nous allons appeler **arp.rules.**

Le programme se place sur la première ligne et lit la ligne courante. La fonction de recherche de motif est appelée pour rechercher le mot « *content* » pour récupérer la signature d'attaque que nous avons spécifié dans la règle. Une fois que nous avons récupéré cette signature, nous le gardons en mémoire.

La fonction recherche de motif est encore appelé, mais cette fois pour rechercher s'il y'a une signature d'attaque dans un paquet. Nous pouvons avoir deux scénarios :

- Si le programme ne retrouve pas la signature dans le paquet et qu'il est arrivé à la fin du fichier (*en testant EOF*), le programme s'arrête et aucune alerte n'est remontée ;
- Si le programme retrouve la signature dans le paquet alors la fonction recherche de motif est encore appelé pour recherche le mot clé *msg.* Apres cette étape le programme récupère le contenu de *msg* et journalise le paquet. Ensuite le programme test *EOF*. Dans le cas où *EOF* est vrai le programme s'arrête, dans le cas où *EOF* est faux, le programme passe à la ligne suivante, et le même procédé est recommencé.

Ce programme est appelé à chaque fois que le décodeur de paquets décode un paquet du protocole ARP.

Ce programme peut aussi être utilisé pour la détection des autres protocoles qui ne sont pas encore pris en compte dans Snort. Comme par exemple le protocole IGMP (*Internet Group Management Protocol),*…etc.

Conclusion Générale et perspectives

Nous avons vu que la sécurisation des réseaux nécessite de prendre en compte plusieurs aspects, qui sont mis en œuvre avec des outils différents. Dans ce travail nous nous sommes intéressés à l'aspect détection des intrusions.

La première partie nous a permis de faire une revue des questions relatives à la sécurité et aux outils qu'il faut utiliser pour assurer la plupart des services de sécurité. A cet effet nous avons étudié les attaques sur la sécurité ainsi que les vulnérabilités dont elles tirent profit. Nous sommes arrivés à la conclusion que la sécurisation des réseaux requiert de prendre en compte plusieurs aspects cryptographiques, organisationnels, protocolaires, physiques, etc.

Dans la seconde partie nous avons étudié les IDS afin d'en connaître le fonctionnement, les contraintes d'utilisation et les problèmes qu'ils soulèvent. Ceci nous a permis de faire ressortir certaines des attaques auxquelles les IDS actuels n'arrivent pas à faire face. Parmi ces attaques nous trouvons celles qui passent par le protocole ARP, et qui ne sont pas détectées par un des IDS les plus utilisés, Snort. Nous avons par conséquent conclu à la nécessité de doter l'IDS Snort d'un mécanisme lui permettant de résister aux attaques de ce type.

Dans la troisième partie une étude plus approfondie de l'architecture et du fonctionnement de Snort nous a permis de cerner les mécanismes utilisés pour détecter les intrusions passant par différents protocoles. Mettant à profit ces connaissances nous avons proposé un algorithme, qui permet à Snort de détecter les intrusions passant par ARP, grâce à la génération de règles spécifiques à ce protocole.

Dans le but d'étudier le comportement des IDS nous avons implémenté l'IDS Snort sur une plate-forme de test que nous avons mise en place. Des simulations d'attaques ont été effectuées pour voir la pertinence des alertes remontées par les IDS. De nombreux problèmes ont été rencontrés, et surmontés, dans cette partie, dûs essentiellement aux difficultés lors de l'installation des différents *rpm* et au nombre important de packages complémentaires à SNORT qu'il faut installer et bien configurer.

Nous nous proposons comme perspective d'implémenter notre algorithme pour le protocole ARP et éventuellement pour les autres protocoles existants non encore pris en compte dans Snort. D'autre part cette approche peut être utilisée pour les attaques via d'autres protocoles.

Une autre perspective concerne l'étude des IDS matériels basés sur l'utilisation des expressions régulières. Ceci est un domaine nouveau dans la détection d'intrusion, car les IDS logiciels montrent leurs limites dés que les débits du réseau dépassent les 100 bits/s [3.14].

Bibliographie

Bibliographie

[1.1] BENCHOUKA SOUFIANE CHERKAOUI Mehdi, Rapport de projet, 2ème semestre 2004-2005

[1.2] Étude de faisabilité IPSec (en coopération avec le DWD, Météo France, le HNMS et le KNMI) : Section réseaux et sécurité Division informatique Mai 2003

[1.3] Honeypot: Des pots de miel pour attirer les pirates, Avril 2007

[1.4] Certificats et infrastructures de gestion de clés, Pierre BARTHELEMY Institut de Mathématiques de Luminy IML - UPR 9016 CNRS Campus de Luminy - Case 907 13288 MARSEILLE Cedex 9 – France Tél. (33)-4-91-26-96-71 / Fax (33)-4-91-26-96-55 barthelemy@iml.univ-mrs.fr

[1.5] International Standards Organisation (ISO). Information Processing Systems - OSI Reference Model. International Standards Organization Publication, Octobre 1984.

[1.7] Signcryption ─BENCHOUKA SOUFIANE─CHERKAOUI : Rapport de projet ─ Projet Signcryption (2ème année) - Cherkaoui & Benchouka 06/06/2005

[1.8] Jean Paul DELAHAYE─ la cryptographie RSA vingt ans après─ Journal « Pour la science N° 267 janvier 2000»

[1.9] L. Mé─ Z. Marrakchi─ C. Michel─ H. Débar ─ F. Cuppens. La détection d'intrusion : les outils doivent coopérer.

[1.10] ICIMAF et Université de la Havane 20 novembre - 1er décembre 2000 La Havane, Cuba

[1.11] B. Chevallier-Mames ─ Cryptographie à clé publique ─Constructions et preuves de sécurité─ THESE─16 Novembre 2006, à l'Ecole normale superieure, Paris pour l'obtention du Doctorat de l'Universit´e Paris VII – Denis Diderot

[2.1] J. Anderson─Computer security threat monitoring and surveillance─1980.

[2.2] D.E.Denning. An intrusion detection model". IEEE Transactions on software engeneering, SE-13:222232, 1987.

[2.3] Nicolas STAMPF avec les contributions de─Maud BRUNSWICK et Raphaël LEONI Quel usage pour les systèmes de détection d'intrusion ? ─ 24/12/2003 disponible à http://stampf.lutin.free.fr/ids/

[2.5] Ludovic Mé et Cédric Miche : La détection d'intrusion, bref aperçu et derniers développements─ l SUP_ELEC B.P. 28 35511─Cesson Sévigné Cedex France─ mars 1999

[2.6] J. H. Holland. "Adaptation in Natural and Artificial Systems". Thèse en Sciences Informatique, University of Michigan, Ann Arbor, 1975.
[2.7] H. Debar, M. Dacier, and A.Wespi─Towards a taxonomy of intrusion-detection systems. internal RZ 3030, ─IBM Zurich Research Laboratory, ─Saumerstrasse 4─ CH-8803 Ruschlikon─Switzerland, June 1998.

[2.8]. Michael Rusinowitch Détection d'intrusion : corrélation d'alertes─Article reçu le 3 décembre 2002─Version révisée le 17 novembre 2003

[2.9] Nathalie Dagorn1 : Rapport de recherche─Détection et prévention d'intrusion─présentation et limites─1 Université de Nancy1─Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA) ─Campus Scientifique BP 239, F-54506 Vandoeuvre-lès-Nancy Cedex, France─nathalie.dagorn@loria.fr─http://www.loria.fr/

[2.10] Marie Barel─Juriste Honeypot : un pot-pourri...juridique─, spécialiste en droit des technologies de l'information et de la communication et sécurité de l'information─marie.barel@legalis.net

[2.11] Ludovic Mé, Zakia Marrakchi, Cédric Michel, Hervé Debar et Frédéric Cuppens. – La détection d'intrusion : les outils doivent coopérer – Revue de l'Electricité et de l'Electronique. pp50-55. No 5, mai 2001.

[2.12] Eric Cariou S─ystèmes Distribués ─ Université de Pau et des Pays de l'Adour Département Informatique

[2.13] Maud BRUNSWICK et Raphaël LEONI : Quel usage pour les systèmes de détection d'intrusion ? ─ Nicolas STAMPF avec les contributions de : ─24/12/2003 ─ Cet article est disponible à l'adresse ─ http://stampf.lutin.free.fr/ids/

[2.14] Ces travaux sont soutenus par la région Bretagne dans le cadre du projet FASTNET.

[2.16] Sophia Antipolis, Marc DACIER, EURECOM, Fabien POUGET, EURECOM ─ Institut de la Francophonie pour l'Informatique─memoire de fin d'etudes─ Programme d'alerte basée sur des pots de miel─ Septembre 2005

[2.17] **Pots de Miel *Honeypots* -**Yann Berthier - Nicolas Jombart-OSSIR Groupe SUR - mars 2002-Herve Schauer Consultants.

[3.1] David Burgermeister, Jonathan Krier : Les systèmes de détection d'intrusion Date de publication : 22/07/2006 ;
Sites : http://dbprog.developpez.com ; http://krierjon.developpez.com

[3.2] Clément Gagnon : Les systèmes de détection d'intrusion─Présentation de, à la journée thématique de l'APVSCI 20 avril 2004 (v2)

[3.3] Patrick Harper | CISSP, RHCT, MCSE with contributions and editing by Nick Oliver Snort, Apache, SSL, PHP, MySQL, and BASE Install on Fedora Core 3 | CNE : http://www.InternetSecurityGuru.com

[3.4] Sylvie Hamel : Motifs - recherche exacte─IFT6291, A2006, Université de Montréal
[3.5] Charot(+), Sylvain Gombault(++), Tony Ramard(++), Christophe Wolinski(+) IRISA Panorama des algorithmes efficaces et architectures, Georges Adouko(+), François, Campus de Beaulieu, 35042 Rennes Cedex_Ces travaux sont soutenus par la r´egion Bretagne dans le cadre du projet FASTNET.

[3.6] Hervé Debar — Benjamin Morin —Frédéric Cuppens ─ Fabien Autrel —Ludovic Mé —Bernard Vivinis RSTI - TSI – 23/2004. Sécurité informatique pages 359 à 390

[3.7] 1998 - 2003 Internet Security Systems, Inc

[3.8] Bro: Un autre IDS Open-Source Jean-Philippe Luiggi Misc Magazine n°14

[3.9] G. Arcas ─ S. Kadhi ─ J.P Luiggi – F. Debieve ─Bro : un NIDS pas comme les autres Réunion du groupe SUR/OSSIR 16 janvier 2007

[3.11] *Guillaume Arcas* : Présentation Prelude IDS 0.9-cvs OSSIR, Support de la conférence de l'OSSIR du 11 juin 2002 http://www.ossir.org/sur/supports/2002/SUR20020611-GArcas.pdf

[3.12] Michael Mullins : Déployer un système de détection d'intrusion, CCNA, MCP, TechRepublic 1 juillet 2002
http://www.freelists.org/archives/helpc/07-2002/msg00049.html

Webographie

[1.6] Didacticiel sur Iptables, version 1.2.0 http://iptables-tutorial.frozentux.net/fr/c96.html

[2.4] IDS - Systèmes de Détection d'Intrusion─Partie I http://www.LinuxFocus.org

[3.10] Antoine AUG : Mise en place de Prelude IDS au sein de votre réseau (SupInfo Paris projet 2005) http://www.supinfo-projects.com/fr/2005/prelude_ids_2005/

GLOSSAIRE

AH: Authentication Header

ARP: Adress Resolution Protocol

CISC: Complex Instruction Set Computer

DMZ: Demilitarized Zone

DoS: Denial of Service

ESP: Encapsulating Security Payload

HIDS: Host Based Intrusion Detection Systems

IDS: Intrusion Detection System

IGMP : Internet Group Management Protocol

ICMP: Internet Control Message Protocol

IETF: Internet Engineering Task Force

IP: Internet Protocol

IPS: Intrusion Prevention System

IPSec: IP Security Protocol

ISS: Internet Security System

LAN: Local Area Network

MAC : Media Access Control

MD5 : Message Digest 5

MiM: Man in the Middle

NAT: Network Address Translation

NIDS: Network Detection Intrusion System

OSI: Open Systems Interconnection

RFC: Request for Comment

RISC: Reduced Instruction Set Computer

SHA: Secure Hash Algorithm

SNORT: The Open Source Network Intrusion Detection System.

TCP: Transport Control protocol

UDP (User Datagram Protocol)

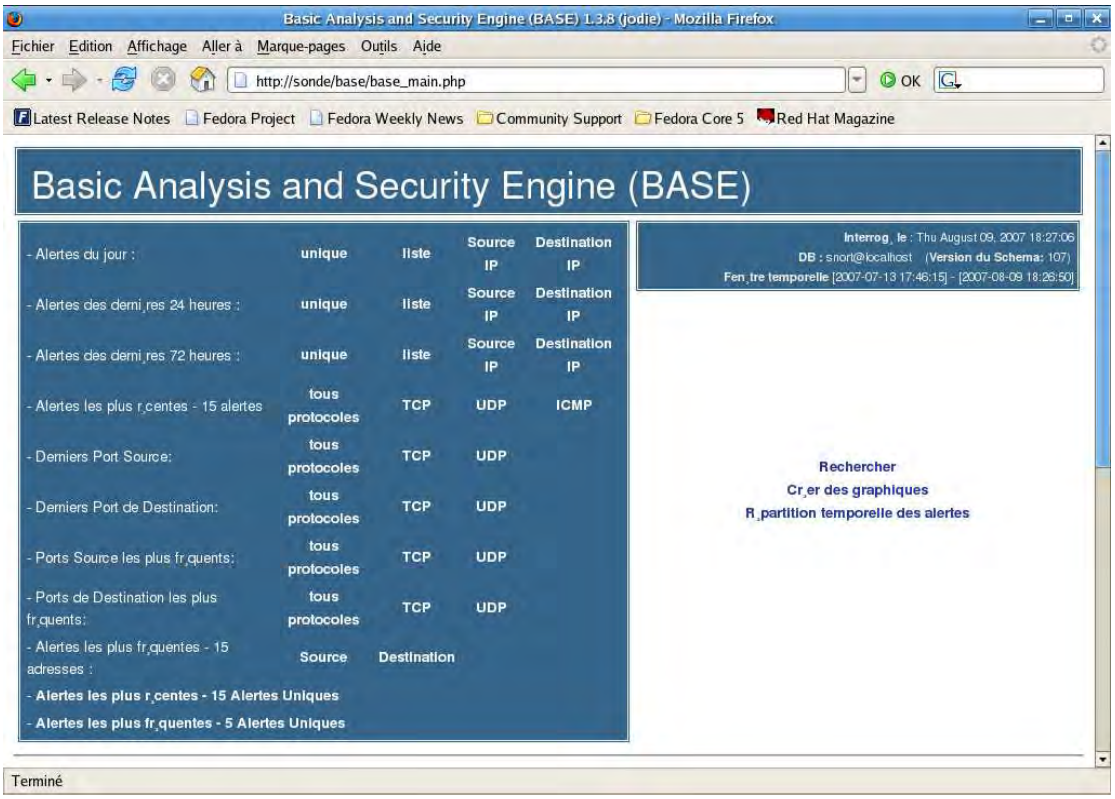
VPN: Virtual Private Network

Annexe

Annexe A : Installation complète de SNORT

```
[root@sonde] # cd /usr/local/src
[root@sonde] # tar –xvzf snort-2.6.1.4.tar.gz
[root@sonde src]# cd snort-2.6.1.4
[root@sonde snort-2.6.1.4]# ./configure --with-mysql=/usr
[root@sonde snort-2.6.1.4]# make
[root@sonde snort-2.6.1.4]# make install
[root@sonde snort-2.6.1.4]# mv /usr/local/man/man8/snort.8
/usr/share/man/man8/
[root@sonde snort-2.6.1.4]# mv /usr/local/bin/snort
/usr/sbin/
[root@sonde snort-2.6.1.4]# mkdir /etc/snort
[root@sonde snort-2.6.1.4]#  CP  -pauvfr etc/* /etc/snort/
[root@sonde snort-2.6.1.4]# mv rules/ /etc/snort/
[root@sonde snort-2.6.1.4]# mv rpm/snort.sysconfig
/etc/sysconfig/snort
[root@sonde snort-2.6.1.4]# mv rpm/snortd /etc/rc.d/init.d/
[root@sonde snort-2.6.1.4]# chmod +x /etc/rc.d/init.d/snortd
[root@sonde snort-2.6.1.4]# chkconfig --add snortd
[root@sonde snort-2.6.1.4]# chkconfig --level 235 snortd on
```

Annexe B : Interface d'accueil de base



Annexe C : vérification des tables créer dans la base de données

```
$]# mysql –u snort –p
>show databases.
+------------+
| Database
+------------+
| mysql
| Snort
| test
+------------+
3 rows in set (0.00 sec)
Pour voir les tables
> use snort ;
> show tables ; // affiche la liste des tables créées
```

Annexe D : configuration de base

```
#$BASE_Language = "english"; # modifier la langue
$BASE_Language = "french";
/* $BASE_urlpath = ""; */
$BASE_urlpath = "../base";
/* $DBlib_path = ""; */
$DBlib_path = "/var/www/adodb";
# Alert DB connection parameters #
#$alert_dbname = "snort_log";
$alert_dbname = "snort";
$alert_host = "localhost";
$alert_port = "";
$alert_user = "snort";
$alert_password = "snort1passe";
# /* Archive DB connection parameters */
$archive_dbname = "snort";
$archive_host = "localhost";
$archive_port = "";
$archive_user = "snort";
$archive_password = "snort1passe ";
```

Annexe E : définition des règles iptables de notre passerelle

```
#!/bin/sh
#*****************************************************************
#** Script de démarrage du firewall*******************************
#**Tout t'es désactiver*****************************************************
#*****************************************************************

#*****************************************************************
#**************Début du script de firewall*************************
#*****************************************************************
inet=eth0
ilan=eth1
```

```
#Les modules : Insertion des modules de traces de connexion (pas nécessaire si intégrés au noyau
#************************ début du chargement des modules
modprobe ip_conntrack_ftp
modprobe ip_tables
modprobe ip_nat_ftp
modprobe ip_conntrack_ftp
modprobe ip_nat_irc
modprobe iptable_filter
modprobe iptable_mangle
modprobe iptable_nat
modprobe ip_conntrack

#********************** fin chargement modules
# Initialise la table Filtrer (par défaut tout les échanges sont refusés)
iptables -t filter -F

# On supprime d'éventuelles chaînes personnelles
iptables -t filter -X

# Mise en place des règles par défaut (on accepte les paquets générés localement)
iptables -t filter -P INPUT DROP
iptables -t filter -P FORWARD ACCEPT
iptables -t filter -P OUTPUT ACCEPT

# Initialise la table NAT
iptables -t nat -F

# On supprime d'éventuelles chaînes personnelles
iptables -t nat -X

# Mise en place des règles par défaut (par défaut tout les échanges sont activés)
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

###############################################################################

# Initialise la table MANGLE
iptables -t mangle -F

# On supprime d'éventuelles chaînes personnelles
iptables -t mangle -X
# Mise en place des règles par défaut (par défaut toutes les modifications de paquets sont autorisées)
iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P INPUT ACCEPT
iptables -t mangle -P OUTPUT ACCEPT
iptables -t mangle -P FORWARD ACCEPT
iptables -t mangle -P POSTROUTING ACCEPT
```

```
#Définir les règles par défaut

#Pour accepter tout ce qui se passe dans l'interface lo dont l'adresse ip est 127.0.0.1

# Autoriser les paquets pour lo
iptables -A INPUT  -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

#Pour accepter tout ce qui se passe sur le LAN

iptables -A INPUT -s 192.168.5.0/24 -j ACCEPT
iptables -A FORWARD -s 192.168.5.0/24 -j ACCEPT
iptables -A FORWARD -d 192.168.5.0/24 -j ACCEPT

#Accepter les connexions http en entrée de l'interface eth0

iptables -A INPUT -i eth0 -p tcp --sport www -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

#Accéder au serveur web depuis l'extérieur

iptables -A INPUT -i eth0 --protocol tcp --sport 80 -m state --state ESTABLISHED -j
ACCEPT
iptables -A OUTPUT -o eth0 --protocol tcp --dport 80 -m state --state
NEW,ESTABLISHED -j ACCEPT

#Accepter les connexions ssh

#on accepte en entrée uniquement si une connexion est déjà établie
iptables -A INPUT -i eth0 -p tcp --sport ssh -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

#on accepte en entrée et en sortie dans tous les cas
iptables -A INPUT -p tcp --dport ssh -i eth0 -j ACCEPT
iptables -A OUTPUT -p tcp --sport ssh -o eth0 -j ACCEPT
#Accepter les connexions FTP

iptables -A INPUT -i eth0 -p tcp --sport 21 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --dport 21 -m state --state NEW,ESTABLISHED -j
ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 20 -m state --state ESTABLISHED,RELATED -j
ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 1024:65535 --dport 1024:65535 -m state --state
ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 1024:65535 --dport 1024:65535 -m state --state
ESTABLISHED,RELATED -j ACCEPT

#se protéger du scan de port utiliser par les hachers pour obtenir des informations sur un
réseau

iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j
ACCEPT
```

```
#se protéger contre le ping de la mort

iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT

#Autoriser le ping, mais dans la limite de 2 pings par seconde (-limit 2/s)
iptables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED,RELATED -j
ACCEPT
iptables -A INPUT  -p icmp -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT  -p icmp -m limit --limit 2/s -j ACCEPT

#Autoriser le ping sans restriction
iptables -A INPUT -p icmp -j ACCEPT

#Autoriser le protocole IGMP

iptables -A INPUT -p igmp -j ACCEPT

#Autoriser les connexions tcp et udp déjà établies à rentrer

iptables -A INPUT -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT

#Activer le Masquerading (Translation d'adresses)

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

#Restreindre le masquerading à une plage d'IPs du réseau local
iptables -t nat -A POSTROUTING -s 192.168.5.0/255.255.255.0 -o eth1 -j
MASQUERADE

#Même règle pour le masque du sous réseau
iptables -t nat -A POSTROUTING -s 192.168.5.0/24 -o eth1 -j MASQUERADE
#Optimisation de la taille des paquets
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS -o eth0 --clamp-mss-
to-pmtu

# A placer en FIN DU FICHIER
iptables -A INPUT -j DROP
```