Sommaire

Remerciements

A Dieu le tout puissant.

Une pensée toute particulière à mes parents et à mes frères et sœurs pour leur soutien de tout instants.

Un remerciement tout particulier à :

Pr. Mamadou Sangharé *Chef du département mathématiques et informatique pour la réussite du master professionnel TDSI.*

A toutes l'équipe pédagogique du master TDSI en particulier **Dr Omar Diankha *et* Dr Cheikh Thiecoumba Gueye.**

M. Djiby Sow *responsable pédagogique MTDSI pour sa relecture et ses idées combien éclairées qui nous sont permit de réussir ce travail.*

Dr. Ibrahim Niang qui est le *responsable de la section informatique.*

Dr. Mbaye Sene *avec qui j'ai découvert le système d'exploitation linux en licence professionnelle et pour avoir accepté de présider mon jury.*

Dr.Mohamed O. Deye *mon encadreur pour son soutien.*

Dr Cheikh Tiecoumba Gueye et Mr Mohamet Sall

A tout mes Amis et collègues de formation pour avoir accepter de participer..

Tous mes remerciements à ceux que j'aurais pu citer.

Introduction

L'utilisation des outils informatique tel que la messagerie instantanée, les communications haut débit, les pages web, les bases de données ouvertes a ouvert la porte à de nouveaux types de criminalités.

Le hacking longtemps ignoré par les responsables informatiques représente maintenant une menace et nombreuses sont les attaques visant à détruire ou à usurper des informations.

Aujourd'hui le premier dispositif de sécurité que l'on trouve dans les réseaux informatique est le firewall, un outil qui permet entre autres de partager un réseau en plusieurs zones (DMZ,LAN…) et d'appliquer des règles de filtrages sur les paquets selon les adresse IP .Mais cet outil se trouve aussi exposer, de ce fait vulnérable aux attaques, alors une fois que l'on peut trouver un autre dispositif qui permet de protéger le firewall cela peut être intéressant .C'est a cela que servent les IDS (Systèmes de Détections d'Intrusions) entre autres.

Le but du mémoire et de faire une étude comparative des systèmes de détection d'intrusion Snort et Prelude-NIDS.

Dans une première partie, nous rappellerons les généralités sur la sécurité des systèmes d'information et sur les systèmes de détections d'intrusions (IDS).
En seconde partie nous ferons une étude comparative des deux IDS Snort et Prelude
Nous terminerons par une troisième partie qui consiste à la mise en place des deux outils dans une plate forme de tests.

Première partie
Généralités

1. Quelques notions de sécurité

De nos jours, le commerce électronique, les opérations bancaires en ligne et une connectivité globale sont les composants essentiels de toute stratégie commerciale gagnante. Pour cette raison, les entreprises ont adopté des processus et pratiques de sécurité afin de protéger leurs informations clés.
La plupart des sociétés prennent un soin particulier à maintenir une stratégie de sécurité performante en implémentant les tout derniers produits et services pour prévenir les fraudes, les actes de vandalisme ou de sabotage et les attaques par déni de service.
Un grand nombre d'entre elles négligent toutefois l'un des principaux éléments d'une stratégie de sécurité réussi qui consiste à définir une bonne politique de sécurité.
Une bonne politique de sécurité doit préserver les aspects de :

–**Disponibilité :** c'est-à-dire fournir l'accès à l'information pour que les utilisateurs autorisés puissent la lire ou la modifier, faire en sorte qu'aucune personne ne puisse empêcher les utilisateurs autorisés d'accéder à l'information.
La disponibilité repose la plus part du temps sur la création d'une redondance entre équipement, sur la mise en œuvre de fonctions de reprise dans les logiciels et sur des procédure de sauvegarde/restitution des données fiable et régulièrement testées.

– **Confidentialité :** c'est-à-dire empêcher les utilisateurs de lire une information confidentielle (sauf s'ils y sont autorisés), empêcher les utilisateurs autorisés à lire une information, de la divulguer à d'autres utilisateurs sans autorisation.

– Intégrité : c'est-à-dire empêcher une modification (création, mise à jour, ou destruction) indue de l'information. Ou encore faire en sorte qu'aucun utilisateur ne puisse empêcher la modification légitime de l'information.
Il est important de s'assurer que la totalité des données à échanger a réellement été échangé, que les données reçues sont bien identiques aux données émises, qu'elles n'ont pas été modifiées lors d'un stockage .Des mécanismes de signature seront alors mis en œuvre ils permettent de mémoriser un état donné et de vérifier que l'état actuel est toujours identique à l'état signé.

– Authentification des utilisateurs et des ressources : pour protéger les informations sensible il convient de savoir qui a le droit d'accéder aux données et quels sont les privilèges associés à ces droits. En l'absence d'une gestion correcte des droits d'accès à l'information, l'entreprise prend le risque de voir des informations confidentielles divulguées, avec toutes les conséquences imaginables.
La détermination des privilèges passe par un mécanisme d'identification et d'authentification qui va permettre de s'assurer que l'utilisateur ou la ressource est bien celui qu'il prétend être ,et qu'il a bien le droit d'accéder au système .le mécanisme peut être simple(login mot de passe) ou fort(certificats , biométrique).

– La non répudiation des transactions : lorsque l'on met en place un service de commerce électronique ou une application transactionnelle sensible il est essentiel de fournir aux deux partenaires qui communiquent le moyen de prouver que l'autre a bien effectué une transaction, de manière à pouvoir réagir si l'interlocuteur prétend ne pas avoir effectué la dite transaction .c'est ce qu'on appelle la non répudiation.
Elle repose sur les mécanismes d'authentification et de signature .Elle introduit aussi les notions de tiers de confiance.

Synthèse des cinq problématiques et les solutions pour y faire face

	Problématique	Mécanisme	Solution
Confidentialité	Protéger des données sensibles	Chiffrement, mécanisme anti-intrusion	Boîtier ou logiciel de chiffrement, pare-feu,IDS
Authentification	Gérer correctement les accès aux données sensible	Mécanisme d'authentification simple ou forte	Login/mot de passe(authentification simple) signature électronique
Intégrité	S'assurer de la non modification des données	Signature,hash-coding	Boîtier ou logiciel de chiffrement (**IPSEC, PGP**) solutions logicielles avec éventuellement carte à puce
Disponibilité	Faire en sorte que les données soient accessibles	Routage dynamique, écrêtage de flux	Redondance des équipements, plan de reprise anti-virus, filtre, sauvegardes
Non répudiation	S'assurer de la réalité d'une transaction	Mécanisme d'authentification, de signature	Solutions logicielles, avec éventuellement carte à puce

2 Les IDS ou Systèmes de Détection d'Intrusion

2.1. Problématique

La détection d'intrusion a pour objectif de détecter toute violation de la politique de sécurité sur un système informatique. Elle permet ainsi de signaler les attaques (en temps réel ou en différé) portant atteinte à la sécurité de ce système, notamment les attaques qui visent des Firewalls ou autres dispositifs de sécurité.

– Par sécurité du système, nous considérons l'intégrité, la confidentialité et la disponibilité du système et des données.

– Par attaque, nous ne considérons pas seulement les intrusions ou tentatives d'intrusion mais aussi d'autres actions telles que les scans, dénis de service , utilisations non autorisées de systèmes/services, mauvaises utilisations de systèmes/services...

Pour mettre en oeuvre ce concept de détection d'intrusion, des outils spécifiques sont nécessaires : les IDS.
Ils vont permettre de collecter de façon automatisée les données représentant l'activité des systèmes (serveurs, applications, systèmes, réseaux), de les analyser et d'avertir les administrateurs en cas de détection de signes d'attaques, ils peuvent aussi générer des contre attaques en bloquant certains ports ou en mettant fin a un trafic suspect.

Dans ce mémoire nous allons dans un premier temps présenter une généralité sur la sécurités informatique et les systèmes de détection d'intrusions.
Ensuite sous feront une études sur la base des études existant déjà des deux outils de détections notamment Snort et Prelude nous terminerons par la mise en place d'une plate forme de test .

2.2 Critères de classification des IDS

Les principaux critères de classifications sont les suivants :

- les méthodes d'analyses
- les sources de données à analyser (réseau/système/application),
- le comportement de l'IDS après intrusion (passif/actif),
- la fréquence d'utilisation (périodique/continue).

Le premier critère de classification des IDS reste la méthode d'analyse. Deux approches sont possibles:

2.2.1 L'approche par scénario

Cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Ce type d'IDS est purement réactif ; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite

des mises à jour fréquentes. De plus, l'efficacité de ce système de détection dépend fortement de la précision de sa base de signature.

2.2.2 L'approche comportementale

Elle consiste à détecter des anomalies. La mise en oeuvre comprend toujours une phase d'apprentissage au cours de laquelle les IDS vont "découvrir" le fonctionnement "normal" des éléments surveillés. Ils sont ainsi en mesure de signaler les divergences par rapport au fonctionnement de référence. Les modèles comportementaux peuvent être élaborés à partir d'analyses statistiques. Ils présentent l'avantage de détecter des nouveaux types d'attaques. Cependant, de fréquents ajustements sont nécessaires afin de faire évoluer le modèle de référence de sorte qu'il reflète l'activité normale des utilisateurs et réduire le nombre de fausses alertes générées.
Chacune de ces deux approches peut conduire à des faux positifs (détection d'attaque en absence d'attaque) ou à des faux négatifs (absence de détection en présence d'attaque).

2.3 Les différents types d'IDS

Un IDS est une machine (sonde) dont la carte est configurée en mode « espion » il a pour fonction d'analyser en temps réel les évènements en provenance des différents systèmes, de détecter et de prévenir en cas d'attaque.
Les buts sont nombreux :

– collecter des informations sur les intrusions,
– gestion centralisée des alertes,
– effectuer un premier diagnostic sur la nature de l'attaque permettant une réponse rapide et efficace,
– réagir activement à l'attaque pour la ralentir ou la stopper.

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

– le trafic réseau : on parle d'IDS réseau ou NIDS (Network based IDS)
– l'activité des machines : on parle d'IDS Système ou HIDS (Host based IDS)
– une application particulière sur la machine : on parle d'IDS Application (Application based IDS).

Contrairement aux deux IDS précédents, ils sont rares. Nous ne les traiterons donc pas.

2.3.1 Les NIDS (Systèmes de Detection d'Intrusions Réseau)

Ces outils analysent le trafic réseau ; ils comportent généralement une sonde qui "écoute" sur le segment de réseau à surveiller et un moteur qui réalise l'analyse du trafic afin de détecter les signatures d'attaques ou les divergences face au modèle de référence. Les IDS Réseaux à base de signatures sont confrontés actuellement à deux problèmes majeurs qui sont :
L'utilisation grandissante du cryptage, et des réseaux commutés. En effet, il est d'une part plus difficile " d'écouter " sur les réseaux commutés (les paquets sont coupés en

plusieurs morceaux) et le cryptage rend l'analyse du contenu des paquets presque impossible.

2.3.2 Les HIDS (Systèmes de Detection d'intrusion sur hot)

Les HIDS analysent quant à eux le fonctionnement ou l'état des machines sur lesquelles ils sont installés afin de détecter les attaques.
Pour cela ils auront pour mission d'analyser les journaux systèmes, de contrôler l'accès aux appels systèmes, de vérifier l'intégrité des systèmes de fichiers ... Ils sont très dépendants du système sur lequel ils sont installés. Il faut donc des outils spécifiques en fonction des systèmes déployés.
Ces IDS peuvent s'appuyer sur des fonctionnalités d'audit propres ou non au système d'exploitation, pour en vérifier l'intégrité, et générer des alertes.

2.3.3 Les IDS hybrides (NIDS+HIDS)

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

3. Utilisation des sondes dans une d'architectures réseaux

Les premières considérations dans les emplacements des NIDS concernaient essentiellement leur position par rapport au(x) firewall(s). S'il est aujourd'hui un élément qui a su trouver sa place dans le système d'information des entreprises, c'est bien le Firewall, parfois même de façon excessive. Cet élément de la chaîne de liaison sécurité est communément placé à divers points stratégiques du réseau pour cloisonner les activités. On trouve ainsi des firewalls entre la bureautique et la production, des firewalls entre une entreprise et ses partenaires, etc.
Le NIDS quant à lui nous délivre des informations en surveillant les le réseau. Il peut aussi se noyer dans la masse et nous signaler un agissement étrange dans une foule.
Plus objectivement, on pourra constater que le firewall permet de séparer des domaines d'activités (fonctionnelles, organisationnelles ou techniques), alors que le NIDS trouvera lui plus facilement sa place dans un domaine de valeur, là ou l'on estime qu'il est intéressant de surveiller ce qu'il se passe exemple la DMZ.
Pour positionner les sondes sur un réseau, quelque soit les besoins.

Nous allons ici présenter 4 solutions :

- **Suivi des attaques :**
- **Surveillance simple** :
- **Surveillance ciblée :**
- **Surveillance multi sites, multi responsabilités :**

(Nous allons détailler des différentes solutions dans ce mémoire)
Mais bien que ces scénarios soient beaucoup plus élaborés que ce que l'on nous donne d'habitude dans les livres , il faudra garder à l'esprit que le meilleur scénario est celui qui sera protégé des différentes zones sensibles du réseau

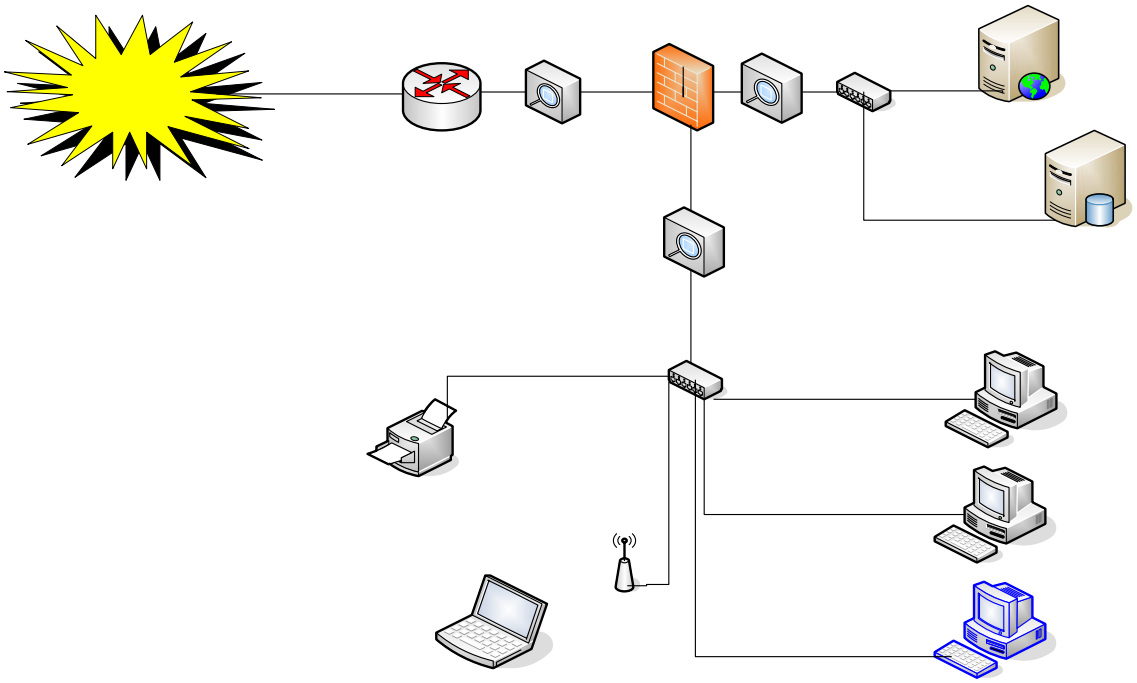


Figure 1 Exemple d'architecture réseaux.

3.1 IDS et DMZ

Super !

Une DMZ représentera une zone d'un réseau pour laquelle on a une confiance moyenne voire faible en raison de son exposition publique. On pourra distinguer la DMZ publique et la DMZ privée, la DMZ publique étant théoriquement une zone où la confiance et la valeur sont inférieures à celle de la DMZ privée. L'utilisation de NIDS en DMZ présente un énorme avantage puisque les flux réseau traversant cette zone sont validés et bien définis. On peut ainsi mettre en place une configuration NIDS très précise, ce qui assure une très bonne pertinence des alertes.
Un environnement avec de nombreuses DMZ nécessitera la mise en œuvre d'autant de sondes.

Routeur

INTERNET

3.2 IDS et LAN

L'utilisation d'un NIDS sur un réseau local est peu répandue, en effet, on peut estimer que la zone est de confiance moyenne, mais surtout que peu de valeur réside sur le

LAN. Néanmoins, on retrouvera sur le LAN des serveurs méritant une attention particulière (DNS, DHCP, serveurs de fichiers, contrôleurs de domaines) ainsi que des machines d'administration, intéressantes pour un rebond ciblé. Un environnement qui respecte les principes de cloisonnement des activités et l'utilisation de VLANs dédiés sera plus facilement utilisable pour déployer des NIDS.
En poussant plus loin l'analyse, il est possible de faire une différenciation entre un LAN de production et un LAN bureautique. Un LAN de production est une zone avec une valeur forte et une configuration correcte. A contrario, un LAN bureautique représente peu de valeur et une confiance moyenne. Dans les deux cas, la configuration NIDS est difficile car on ne maîtrise pas souvent les flux qui transitent sur le réseau interne et les ressources sont souvent éparpillées. Pour le LAN, nous pouvons parler des cœurs de réseau, qui concentrent les flux venant de VLANs différents. L'avantage des cœurs de réseau pour y placer une sonde est une visibilité accrue sur tous les VLANs, mais le principal désavantage réside dans la quantité de trafic. Les nouveaux cœurs de réseau sont maintenant prévus en dizaine de gigabits dans les grandes entreprises et même si des produits se déclarent « Gigabit compliant », leur débit utile est souvent de l'ordre de quelques centaines de mégabits.
De plus, si les paquets sont de petite taille, la capacité de traitement s'effondre.
Une sonde aurait donc plus d'intérêt à être située sur un équipement d'accès au cœur de réseau (routeur, Firewall…) ou sur la partie distribution

3.3 IDS et MAN

Un MAN permet d'interconnecter des sites géographiquement proches. Les sites peuvent hébergés la production de l'entreprise, les serveurs bureautiques, être des sites bureautiques ou mélanger ces possibilités. Sur le MAN, nous pouvons donc voir tout type de flux (bureautique/ production). Si la séparation des activités est faite, il est facile d'identifier le trafic passant sur les liens du MAN. Dans ce cas, les flux non identifiés seront suspects. Dans les deux cas, l'utilisation d'un NIDS est intéressante puisqu'elle permettra d'analyser les flux à l'interconnexion des sites. Comme énoncé dans le paragraphe précédent, mettre une sonde à l'entrée d'un site permet d'analyser tout ce qui rentre et sort de ce site. Qu'il y ait une politique de filtrage entre les sites et LAN de chaque site ou non, le NIDS permettra une analyse plus profonde des flux (Rappelons qu'un flux autorisé par un firewall/ACL n'est pas forcément valide).
Rappelons aussi que les liens réseaux du MAN étant dans la plupart des cas des liaisons louées, ils peuvent être partagés entre plusieurs clients. Nous pourrions affecter une note de valeur de faible à moyenne, et une note de confiance qui sera correcte si le lien ne supporte qu'un client (fibre entre deux bâtiments) ou plus forte si la liaison est partagée. Cela peut amener à se poser la question du chiffrement des flux. Ceci implique le placement de la sonde aux endroits où le trafic n'est pas chiffré.

3.4 IDS et WAN

Les problématiques du WAN ressemblent à celles du MAN. La charge réseau qui transite par le WAN est généralement plus restreinte et plus précise que celle qui passe

par le LAN car les lignes sont souvent louées. Ainsi, un WAN obtiendra une note de confiance correcte car les flux réseaux sont maîtrisés, mais à l'inverse la zone de valeur sera probablement plus faible que pour un MAN.

4. Suivi des attaques

Les IDS sont rarement les seules solutions de sécurité informatiques mises en place sur un réseau. Il y a donc tout un ensemble de "barrières" déjà mises en place. Mais sont-elles efficaces ? Et si une attaque est perpétrée contre une machine du réseau, quelle protection a permis de la stopper, et laquelle n'a au contraire rien changer ? Un réseau peut aussi avoir des vulnérabilités de conception, comme par exemple, la possibilité de se connecter à des Access Point Wireless, accédant de la sorte au LAN sans passer par le firewall qui ne surveille que les accès à Internet.

4.1Surveillance globale

La mise en place de sondes sur tout le réseau va permettre de suivre une attaque. Il est important de pouvoir avoir une sonde surveillant le traffic avant et après chaque protection (avant et après un firewall...). Ainsi, cela permet d'avoir un aperçu des attaques que la protection bloque et celles qu'elle laisse passer.
Pour une attaque donnée, on pourra en déduire son *taux de pénétration* dans le réseau, et ainsi désigner les parties du réseau sûres et celles non sûres face à une attaque donnée.
Dans les réseaux complexes et vastes, nous nous trouvons parfois face à des niveaux de sécurité hétérogènes. Le cas le plus répandu est de rencontrer une sécurité maximale sur l'accès à Internet, mais une fois dans le LAN, tout est ouvert (faible protection entre la DMZ et le LAN, des Access Point Wireless poussant comme des champignons, accès illicites à Internet directement d'un poste de travail, ...). Alors qu'une attaque donnée serait bloquée en passant par la grande porte, elle ne rencontrera aucune résistance en passant par un autre chemin. Le positionnement des sondes sur tout le réseau va alors permettre de détecter par où elle est rentrée, ou le chemin qu'elle a emprunté pour contourner les protections qui auraient pu la bloquer.

Avantages/Inconvénients :
Le problème de cette solution est le coût matériel et humain. Le nombre de sondes étant très important, il est évident que le nombre de machines à dédier à cette tâche est plus important. De plus, l'exploitation correcte de toutes les alertes remontées (alertes remontées par plusieurs sondes, savoir retrouver le parcours d'une attaque dans tout cela, ...) nécessite beaucoup de temps.

4.2 Surveillance simple

Il est ici question de détecter une attaque. Le chemin suivi par cette dernière n'est plus important. Les sondes sont placées seulement sur les points d'entrées des réseaux et absentes du *backbone (cote Internet).*
Alors que précédemment les sondes devaient toutes avoir activées les mêmes signatures d'attaques pour le suivi, nous pouvons ici faire un paramétrage plus fin. Dans un réseau A sous Windows (IIS), on va activer les alertes Unicode, mais sous un réseau B, non. Nous évitons de la sorte les remontées d'alertes pour des attaques dont la cible est invulnérable. L'administration des IDS s'en trouve tout autant simplifiée.

Avantages/Inconvénients :
Contrairement au cas précédent, le nombre de sondes est beaucoup plus réduit. Étant donné que nous recherchons à détecter les attaques dont nous sommes vulnérable, nous pouvons alléger les remontées d'alertes. Cette solution est alors beaucoup moins coûteuse que la précédente. Le revers de la médaille est que l'on perd en précision, et qu'il n'y a plus de redondance dans la surveillance (zone surveillée par au moins deux sondes).

4.3 Surveillance ciblée

Jusqu'à présent, le nombre de sondes est important. Cela implique une maintenance lourde et coûteuse sur les grands réseaux. Les points d'un réseau ne sont pas aussi sensibles les uns que les autres. Cela est valable sur les réseaux très vaste : il est préférable de bien surveiller les parties sensibles du réseau que de vouloir tout contrôler avec le même niveau de sécurité ... chose qui est impossible (ou alors coûteux).
La mise en place doit, bien sûr, être précédée d'une phase d'étude pour déterminer quelles sont les zones sensibles, et celles qui le sont moins.

Avantages/Inconvénients :
Le coût est ici très faible. La complexité de la surveillance est réduite, ce qui fait gagner d'autant en efficacité et en rapidité de réaction en cas d'incident de sécurité. Nous comprenons bien que l'efficacité des IDS dépendra beaucoup de l'étude préalable du réseau.

4.4 Surveillance multi-sites, multi-responsabilités

Nous traitons ici le cas particulier des réseaux très étendus, multi-sites, dont la gestion de la sécurité n'est pas entièrement centralisée. Si un incident de sécurité se produit sur le site A, ce serait une perte de temps de remonter l'alerte sur le site C où un responsable sécurité préviendra le responsable sécurité du site A de l'incident. Alors autant faire en sorte que les alertes de chaque sonde soit remontées au manager du site.
La modularité des filtres sur les managers permet même d'aller plus loin. En effet, rien ne nous empêche de positionner plusieurs managers comme dans le site vert, afin

qu'un premier manager n'affiche que les alertes vertes, un autre que les alertes oranges, et ainsi de suite.
Ce partage entre plusieurs managers permet de rediriger les alertes vers différentes personnes, suivant leurs niveaux de compétences.
Nous pourrions rediriger les alertes vertes vers un technicien, les alertes oranges vers un ingénieur réseau, et les alertes rouges vers un ingénieur expert sécurité.
Nous reproduirions alors les niveaux supports, au niveau de la maintenance sécurité : support de niveau 1, puis si le problème persiste nous passons au niveau 2, etc. Ici, il y aurait en plus la possibilité pour une alerte d'arriver directement au niveau 2 ou 3, suivant sa gravité (réseau très sensible à ce type d'attaque, alertes rouges, ...).
Enfin, une dernière possibilité serait de combiner le traitement des alertes par un manager sur chaque site mais, dans le même temps, remonter les alertes vers un autre manager qui centraliserait ainsi les alertes de tout le réseau informatique de l'organisme (pour réaliser un historique par exemple).

Avantages/Inconvénients :

Il n'est plus question ici de surveillance d'un petit réseau, mais de l'intégration des IDS dans une politique de sécurité globale de l'organisme. Les traitements séparés avec plusieurs niveaux d'intervention montrent que la sécurité n'est pas que technique mais aussi organisationnelle.
L'avantage de cette solution est donc de s'intégrer totalement dans une politique de sécurité, ce qui induit qu'il est nécessaire d'avoir une organisation de la sécurité stable, et bien définie.

5. Faiblesses de la méthode de protection par IDS

Les IDS ne sont pas là pour remonter des alertes d'attaques involontaires (faute de frappe, erreur de saisie) mais plutôt pour détecter des attaques plus élaborées. Toute attaque un minimum préparée, comprend une phase de camouflage ou d'effacement des traces. Sur un système, cela passe par l'effacement des logs ou la modification des attributs des fichiers modifiés. Dans le cas de traces réseaux, cela va passer par l'attaque des sentinelles : les IDS.
Mais d'un autre coté, la simple utilisation des IDS pose quelques problèmes que nous allons maintenant détailler.

5.1 Connaissances en sécurité

La mise en place de sonde sécurité fait appel à de bonnes connaissances en sécurité. L'installation en elle-même des logiciels est à la portée de n'importe quel informaticien. En revanche l'exploitation des remontées d'alertes nécessite des connaissances plus pointues. Les interfaces fournissent beaucoup d'informations, et permettent des tris facilitant beaucoup le travail, mais l'intervention humaine est toujours indispensable. A partir des remontées d'alertes, quelle mesure prendre ? Est-il utile de relever des alertes dont toutes les machines sont protégées (attaques sur MSSQL sur un réseau ayant que du MySQL) ? Comment distinguer un faux positif

d’un véritable incident de sécurité ? Par exemple un nombre important de icmp-redirect peut être le signe d’une attaque de type "homme du milieu", mais aussi d’un routage mal configuré.

5.2 Problème de positionnement des sondes

La mise en place est importante. Il faut bien définir là où placer les sondes. Il ne s’agit pas de mettre une sonde partout où l’on veut surveiller. Il faut étudier les champs de vision des sondes suivant leur placement, si on veut recouper ces champs de vision (pour par exemple faire des doublons de surveillance ou faire un suivi d’attaque), quel détail d’analyse (à l’entrée d’un réseau, ou dans chaque domaine de collision). On découpe souvent le réseau global en un LAN, une DMZ, puis Internet. Mais il faut aussi envisager les domaines de collisions, les sous réseaux, ...
Étant donné que la sonde travaille en mode promiscuité, elle utilise la librairie libpacap ou winpcap qui fait qu’une sonde ne pourra pas être installée sur les firewalls. Et la mise en place sur la sonde même d’un filtrage pour la protéger contre certaines attaques directes aura une efficacité très réduite.
L’utilisation de tunnel est aussi à envisager lors du positionnement des sondes : inutile de placer une sonde où tout le trafic est crypté.
Les connaissances réseaux sont importantes. Il faut aussi faire attention à comment sont remontées les alertes (si on passe par une ligne RNIS, éviter de la monter et la fermer à chaque alerte). Même si la plupart des schémas montrent un manager et N sondes, nous pouvons très bien utiliser M manager et N sondes.

5.3 Vulnérabilités des sondes NIDS

De part leur fonctionnement en mode promiscuité (carte réseau sans adresse IP), les sondes sont vulnérables. Elles captent tout le trafic, et même si un ping flood est réalisé sur une autre machine, les sondes NIDS le captureront aussi et donc en subiront les conséquences, comme si l’attaque leur était directement envoyée.

5.4 Un DoS explose les fichiers de logs

Le point fort de certains IDS qui est d’archiver aussi le payload des trames ayant levées une alerte, peut aussi s’avérer un point faible. Un ping flood avec un payload chargé de 64000 octets, ou encore des trames de 1500 octets pour les SYN flood vont faire exploser la taille des fichiers de logs des sondes en quelques minutes. C’est une attaque qui porte le nom *coke* qui consiste à saturer le disque dur (http://www.securiteinfo.com/attaques/hacking/coke.shtml). La seule façon de parer cette attaque est de prévoir d’importants espaces de stockages, et gérer le stockage des fichiers de logs.

5.5 Réponse active non contrôlée

Certains IDS génèrent une contre-attaque lorsqu'ils détectent des attaques. Le gros problème est de la justification de cette contre-attaque. On pourrait suivre le principe du DDoS pour faire attaquer les sondes.
Par exemple, on réalise un ping echo-request-broadcast sur quelques machines, en faisant croire que la source de la demande est une machine du réseau (ou plusieurs si on est très joueur). Toutes les machines vont répondre à cette machine. Ainsi on a un premier niveau de DDoS. Les sondes interviennent alors.
Détectant cette attaque comme tentative de surcharge du réseau, elles contre-attaquent la source du ping.
On a ainsi un second niveau de DDoS encore plus nocif. En plus d'un engorgement réseau, on a réussi à saturer une machine. Il suffit que cette dernière soit un serveur avec une pile IP faible non protégée, et nous pouvons dire qu'il est planté.
Imaginons maintenant que les contre-attaques des sondes sont vues par l'attaquant (il met son interface en mode promiscuité ou ne camoufle pas le fait qu'il est la source de l'attaque... il préfèrera alors un simple ping flood pour ne pas subir le premier niveau de DDoS). Il va alors pouvoir détecter la présence de sondes IDS, leur nombre, et leur type. Ainsi, il connaît avec certitude une partie des mesures de sécurité prises sur tout le parcours de l'attaque du réseau cible ... choses que les responsables sécurité n'aiment pas divulguer.
Nous rappelons, comme expliqué précédemment, les problèmes d'explosion des fichiers de logs que ces DDoS entraînent.
Voici un dernier point qui utilise le revers de la médaille des contre-attaques automatiques : si nous mettons 2 sondes sur 2 réseaux séparés A et B, et qu'une attaque se produit sur le réseau B en passant par le réseau A.
La contre-attaque de la sonde du réseau B sera vue par la sonde du réseau A. Cette dernière va donc contre-attaquer la source de la première attaque (si on avait été joueur, on aurait falsifié la source en indiquant la sonde du réseau C, réseau qui se trouve derrière B), mais aussi la source de la deuxième attaque qu'est la sonde B.

5.6 Problèmes de IPv4

Comme détaillé plus tard dans ce mémoire, il est facile de contourner certains IDS en utilisant les fragments IP. On exploite ici une faiblesse du protocole IPv4, et des piles IP.
Une autre faiblesse de ce protocole est de ne pas permettre l'authentification. On peut donc faire croire à un destinataire qu'un paquet vient de telle ou telle source alors que cela serait faux. Cela est la base des DoS et DDoS évoqués précédemment.

5.7 Problèmes intrinsèques à la plateforme

Beaucoup d'IDS (et plus particulièrement les IDS libres) sont des logiciels reposant sur un système d'exploitation non dédié aux IDS. Ainsi, la faiblesse d'un IDS est liée à la faiblesse de la plateforme.

Un même logiciel sera par exemple plus vulnérable sur un PC Win98 que sur un PC OpenBSD, de part la solidité de la pile IP face aux attaques, ou tout simplement de part la stabilité du système. La mise en place d'un IDS requiert donc des compétences dans la sécurisation de la plateforme.
Une saturation de la mémoire, de la carte réseau (éviter des cartes ISA), ou du processeur porte atteinte directement au bon fonctionnement de tout le système et donc du logiciel IDS de la machine. Le problème de ces dysfonctionnements est que si la sonde ne peut plus remplir son rôle, le réseau n'en est pas coupé pour autant
Le responsable sécurité ne peut donc pas voir que la sonde étant tombée, une partie du réseau n'est plus surveillée.
Une redondance des surveillances sur certaines zones, devraient momentanément résoudre le problème. Mais contre une attaque bien préparée, c'est inutile. Malheureusement, nous n'avons pas eu le temps de tester tout cela, afin de connaître les comportements des IDS à leurs limites de fonctionnement.

5.8 Coût matériel

La mise en place d'un IDS peut entraîner des coûts matériels importants. Effectivement, sur un grand réseau, le nombre de sondes et de managers peut être important, et donc le nombre de PC ou "de boîtes IDS propriétaires" aussi. Encore une fois, l'utilisation de logiciels libres ou open-sources permet de beaucoup réduire ce problème.

6. Méthodes de contournement des IDS

Les systèmes de détection d'intrusion, aussi performants qu'ils puissent être présentent certaines limites(bases de signatures obsolètes, faux positifs, faux négatifs ...).Ces limites peuvent être utilisées pour attaquer ou passer au travers des IDS.

6.1 Quelques techniques

Il existe ainsi plusieurs techniques pour échapper à la détection par les IDS notamment dans le cas des IDS par scénario :

- **L'évasion**

C'est faire passer une attaque au travers du système de détection en évitant de correspondre à un scénario répertorié, donc détectable.

- **L'insertion**

C'est l'insertion de trafic qui permet de déjouer l'IDS en lui faisant croire à un trafic légitime. On injecte l'attaque parmi beaucoup d'informations sans incidences. Les signes de l'attaque n'apparaissent donc pas à l'IDS mais quand les données atteignent

la cible, seule l'information malintentionnée est acceptée par le système. D'autres méthodes sont possibles :

- **Denis de service**

- **Flood de signes d'intrusions** (faux positifs ou vraies alarmes) pour déclencher beaucoup d'évènements et surcharger les administrateurs. Nous pouvons par exemple utiliser l'outil **snort** pour cela.
- **Contournement physique** (les trames ne sont pas capturées par les sondes) en jouant sur les domaines de collisions.
- **Attaque directe contre l'IDS** : comme nous l'avons vu dans ce rapport, les IDS présentent quelques points faibles qui peuvent être exploités (réponse active redirigée, saturation de l'espace de stockage des alertes, faiblesse de la pile utilisée, ...)

6.2 Exemples d'attaques

Pour illustrer ces différentes méthodes, voici quelques exemples d'attaques permettant d'outre-passer les IDS :

6.2.1 Les attaques réseaux

Le but principal est de réduire les possibilités du NIDS à détecter les attaques :

- **Par les méthodes classiques de scan :**

Nous pouvons prendre comme exemple le scan furtif SYN implémenté par NMAP permettant de ne pas être détecté par les NIDS. Le but du scan SYN est de ne pas ouvrir une connexion complètement. A la réception d'un SYN/ACK qui signifie que le port est ouvert, il envoie un RST pour interrompre la connexion. Aucune connexion n'est donc faite tout en sachant quels ports sont ouverts.

- **Par le flood :**

Il consiste à surcharger le NIDS pour qu'il ne puisse pas fonctionner correctement, et qu'il ne détecte pas l'attaque principale.
Par la noyade sous les faux positifs : Le principe est de provoquer de nombreuses remontées d'alertes. En parallèle, une attaque réelle contre le réseau est lancée, et l'administrateur occupé à analyser les alertes, ne s'en rendra pas compte sur le moment. Nous pouvons utiliser l'option decoy de nmap pour générer des scans nmap multiples.

- **Par fragmentation :**

Le principe est de fragmenter les paquets IP pour empêcher les NIDS de détecter les attaques sachant que les paquets seront réassemblés au niveau du destinataire. Il est possible aussi d'envoyer des paquets IP mal fragmentés qui vont utiliser la faiblesse de

certaines pile IP (peut-être aussi celle de la sonde IDS qui capte tout le trafic) pour perturber le système.

- **Par des scans lents :**

Les NIDS ne détectent souvent pas ce type de scan (un scan toutes les heures par exemple). Sachant qu'un NIDS maintient un état de l'information (TCP, IP fragments, TCP scan, ...) pendant une période de temps bien définie (capacité mémoire, configuration), ils ne détectent rien si deux scans consécutifs sont trop espacés.

7. Conclusion

Les IDS continuent d'évoluer pour répondre aux exigences technologiques du moment et offrent d'ores et déjà un éventail de fonctionnalités capables de satisfaire les besoins de tous les types d'utilisateurs.
Cependant comme tous les outils techniques ils ont des limites que seule une analyse humaine peut compenser. Les détecteurs d'intrusions deviennent chaque jour meilleur grâce à l'expérience acquise avec le temps. Mais ils deviennent aussi de plus en plus sensibles aux erreurs de configuration et de paramétrage. Par conséquent, il est plus que fondamental de former correctement les personnes chargées de la mise en oeuvre et de l'exploitation des IDS.

Deuxième partie
Etude comparative de Snort et Prelude

1 Snort et Prelude-NIDS
1.1 Snort

Snort est un système de détection d'intrusion réseau capable d'effectuer l'analyse du trafic en temps réel et de la journalisation de paquets sur des réseaux IP. Il peut effectuer de l'analyse de protocoles, de la recherche / correspondance de contenu et peut être utilisé pour détecter une variété d'attaques et de scans, tels que des débordements de tampons, des scans de ports furtifs, des attaques CGI, des scans SMB, des tentatives d'identification d'OS... Snort utilise un langage de règles flexibles pour décrire le trafic qu'il devrait collecter ou laisser passer, ainsi qu'un moteur de détection qui utilise une architecture modulaire de plugin. Snort possède aussi des capacités modulaires d'alertes en temps réel, incorporant des plugins d'alerte et de journalisation pour syslog, des fichiers textes en ASCII, de sockets UNIX, de messages WinPopup à des clients Windows en utilisant smbclient de Samba, de base de données (Mysql/PostgreSQL/Oracle/ODBC) ou XML. Snort a trois utilisations principales.
Il peut être utilisé comme un simple renifleur de paquets comme tcpdump (1), un enregistreur de paquets (utile pour déboguer le trafic réseau, etc), ou comme un système complet de détection d'intrusion réseau.
Snort journalise les paquets dans le format binaire tcpdump, vers une base de données ou dans le format ASCII

1.1.1 Caractéristiques

- **Descriptions**

L'architecture de SNORT est organisée en modules, elle est composée de quatre grands modules : Le décodeur de paquets, les préprocesseurs, le moteur de détection et le système d'alerte et d'enregistrement de log.

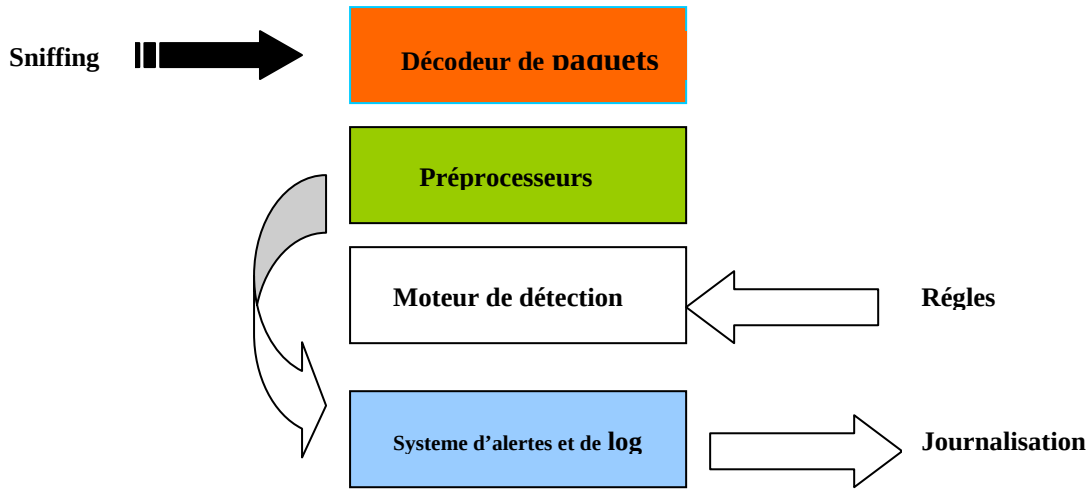


Figure 2. Architecture de Snort

- **Le décodeur de paquets.**

Un système de détection d'intrusion active un ou plusieurs interfaces réseau de la machine en mode espion (promiscuous mode), ceci va lui permet de lire et analyser tous les paquets qui passent par le lien de communication. SNORT utilise la bibliothèque **libpcap** pour faire la capture des trames.
Un décodeur de paquets est composé de plusieurs sous décodeurs qui sont organisés par protocole (Ethernet, IP, TCP..), ces décodeurs transforme les éléments des protocoles en une structure de données interne.

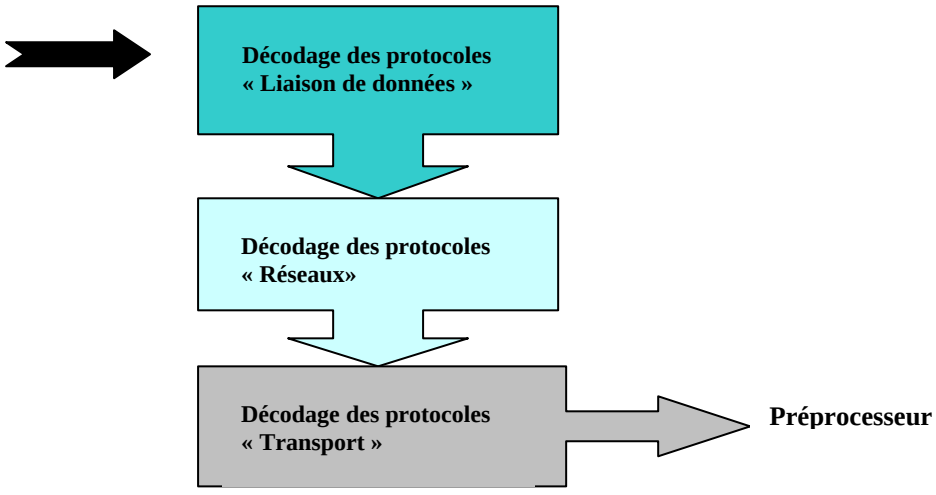


Figure 3. Décodeur de paquets

- **Les présprocesseurs.**

Les préprocesseurs s'occupent de la détection d'intrusion en cherchant les anomalies, un pré processeur envoie une alerte si les paquets ne respectent pas les normes des protocoles utilisées. Un pré processeur est différent d'une règle de détection, il est un programme qui vise à aller plus en détail dans l'analyse de trafic.

- **Moteur de détection.**

C’est la partie la plus importante dans un SDI. Le moteur de détection utilise les règles pour faire la détection des activités d’intrusion. Si un paquet correspond à une règle une alerte est générée.
Les règles sont groupées en plusieurs catégories sous forme de fichiers. SNORT vient avec un ensemble de règles prédéfini.
Ces règles ne sont pas activées automatiquement, il faut les activer dans le fichier de configuration **snort.conf**. Chaque fichier contient des règles décrivant un type de trafic à signaler.

- **Système d’alerte et d’enregistrement des logs.**

Le système d’alerte et d’enregistrement des logs s’occupe de la génération des logs et des alertes. Les alertes sont stockées par défaut dans le répertoire **/var/log/snort/.**
Dés que le système devient opérationnel, on pourra consulter les alertes générées directement dans les fichiers textes ou bien utilisé une console de gestion. ACID (Analysis Console for Intrusion Detection), est une application qui fournie une console de gestion et qui permet la visualisation des alertes en mode graphique. Les alertes dans ce cas sont stockées dans une base de données mysql.

1.1.2 Les codes source des scripts

SNORT est une application écrite en C. les programmes sources sont dans le répertoire **snort-2.1.0/src/.** Le programme **snort.c** représente la routine principale de SNORT, le décodeur des paquets est implémenté dans le programme **decode.c**. **rules.c** est la routine qui s’occupe des règles.
Le moteur de détection est implémenté dans le programme **detect.c** et Le moteur d’enregistrement est dans **log.c**.
Il est très utile de consulter le contenu de ces programmes et voir comment SNORT capture les paquets et détecte les attaques.

1.2 Prelude NIDS

Le projet de Prelude a commencé en 1998 et avait pour but de créer un outil modulaire de détection d'intrusion réseau composé d'une sonde et d'un Report Server. Lors du Libre Software Meeting 2001, les équipes de Prelude et du projet Trithème (projet indépendant lancé en février 2000) ont décidé de joindre leurs efforts dans le but d'évoluer progressivement vers le développement d'un IDS hybride basé sur la prise en compte de la quasi-totalité des évènements sécurité au niveau réseau (Network-based IDS) et local (Host-based IDS) grâce à des sondes dédiées.

1.2.1 Caractéristiques.

- **Description**

Prelude-IDS (Intrusion Detection System) est un système de détection d'intrusions et d'anomalies distribué sous licence GPL.
Un tel système vient compléter la panoplie des équipements et logiciels de sécurité (serveurs proxy ,routeurs filtrants, firewalls...) et offre à l'analyste un outil de contrôle des activités suspectes ou illicites(interne comme externe).
La détection d'intrusion est réalisée par l'analyse du trafic réseau et l'utilisation de signatures d'évènements hostiles ou par l'analyse en continue de fichiers de journalisation.
L'architecture de Prelude est :

--Modulaire (on peut intégrer ou développer de nouvelles fonctionnalités grâce à des plugins).

--Distribuée (Prelude est une suite de composants autonomes et interactifs, ie les sondes et les managers) .

--Sécurisée (utilisation du support SSL pour l'authentification et le chiffrement des communications).

Les sondes (réseaux comme locales) n'effectuent que les opérations de surveillance et de génération d'alertes alors que les managers prennent en charge la gestion des sondes et la journalisation des alertes.

- **Architecture**

– Les capteurs sont des entités de détection capables de remonter des alertes à un manager Prelude.

– Le manager (il peut y en avoir plusieurs) accepte les connexions en provenance des différents capteurs et collecte leurs alertes. Il assure les fonctions suivantes :

Le logging qui permet de transformer une alerte au format Prelude en un format lisible par l'analyste.

La contre-mesure qui permet à l'utilisateur de définir une réaction à une attaque.

– Les agents de contre-mesure sont placés sur les machines devant opérer la réaction à une attaque (fonction en cours de développement).

– Le frontend est une interface d'administration web permettant d'aider les administrateurs sécurité à analyser plus facilement les alertes et les statistiques, sachant que cette tâche ne peut être complètement automatisée à l'heure actuelle.

–La communication entre les différents programmes se fait au format IDMEF (Intrusion Detection Message Exchange Format). Ce format fondé sur XML est suffisamment générique pour permettre aux composants hybrides de Prelude d'émettre des alertes décrivant des événements de tous types : attaques réseau, buffer-overflow local, ...

➢ **Composants**

- **Libprelude (la librairie Prelude)**

La librairie libprelude constitue la brique de base de tout composant Prelude à l'exception du frontend. Cette librairie fournit aux composants Prelude les fonctionnalités suivantes :

– Gestion de la connexion entre composants (sondes et managers) notamment le mécanisme de reprise après interruption et de rétablissement automatique de la connexion ;

– Gestion du mode de communication entre composants, notamment la prise en charge du chiffrement éventuel et de l'authentification ;

–Interface permettant l'intégration de plugins. Cette librairie doit être installée préalablement à l'installation de tout autre composant (à l'exception toujours du frontend).

Cette sonde prend en charge l'analyse en temps réel du traffic réseau.
Elle est construite au-dessus de la librairie **libprelude** et fournit :

– Un moteur de gestion de signatures générique, actuellement compatible avec les signatures Snort, mais pouvant être étendu par l'ajout de nouveau "parser" de règles ;

– Des modules spécialisés par protocole : par exemple, un plugin est dédié aux protocoles RPC et permet l'analyse fine de ce type de connexions ;

– Des modules spécialisés dans la détection non basée sur des signatures : détection des activités de balayage (scan), ...

Les sondes réseaux peuvent aussi prendre en charge la défragmentation IP et le réassemblage des flux TCP.

- **Prelude-LML (la sonde locale)**

Cette sonde prend en charge la remontée d'alertes détectées localement sur une machine. Cette détection est basée sur l'application à des objets (fichiers de journalisation et/ou application) de règles construites autour d'expressions régulières compatibles Perl (PCRE).
Pour la surveillance des systèmes Unix, une sonde prelude-LML peut utiliser le service syslog et ainsi assurer la remontée d'alertes. L'intégration des systèmes Microsoft peut également se faire à l'aide de l'utilitaire ntsyslog.

Un message est généré par la sonde prelude-LML dès qu'une ligne de log correspond à une expression régulière.

- **Prelude-Manager (le contrôleur)**

Prelude-manager centralise les messages des sondes réseaux et locales, et les traduit en alertes. Il est responsable de la centralisation et de la journalisation à travers deux fonctions :

– Celle de relais : un contrôleur relais va assurer le routage vers un contrôleur maître d'alertes provenant des sondes qui lui sont rattachées.

– Celle de maître : un tel contrôleur va assurer la réception des messages et des alertes provenant des sondes ainsi que leur journalisation dans un format lisible par l'analyste : en mode texte (dans les fichiers) ou SQL dans le cas de l'utilisation d'un SGBD (MySQL ou PostgreSQL).

Il est possible d'étendre les capacités d'un contrôleur à l'aide de plugins, en autorisant, par exemple, le traitement de messages en provenance de composants autres que Prelude, un contrôleur Prelude pouvant ainsi centraliser la remontée d'alarmes en provenance **de sondes Snort.**

- **Prelude-Frontend (l'interface web)**

C'est l'interface de visualisation des alertes. Il est actuellement proposé deux interfaces, l'une développée en PHP et l'autre en Perl.

– Prelude-PHP-Frontend C'est l'interface proposée sur le site "www.prelude-ids.org". Elle est composée de scripts PHP et est destinée à être installée sur un serveur web indépendamment des autres composants Prelude. Cela signifie que l'installation préalable de la librairie libprelude est inutile, mais que par contre celle d'un serveur web supportant PHP4 l'est.

.

– Prelude-Perl-Frontend Cette interface est issue d'un projet intitulé "Le Routier"
(www.leroutier.net/Projects/PreludeIDS). Elle nécessite bien évidemment l'installation d'un serveur web supportant Perl.

1.3 Comparaison de Prelude-IDS et Snort

Nous allons étudier les points communs et les différences entre ces deux outils en commençant par évoquer les projets qui les ont fait naître. Puis nous partirons du niveau le plus bas qu'est le moteur d'analyse, pour finir à l'interface de remonter des alertes. Enfin, nous dirons quelques mots sur la facilité de configuration, et la possibilité d'intégration d'outils externes au projet.

- **Projet**

Dans le cadre du projet, ces deux outils sont très proches. Ce sont tous les deux des outils libres. Les projets dont ils sont les résultats sont très actifs, aussi bien dans le développement que dans la mise à jour des attaques.

Quelques avantages de Snort sur Prelude-IDS sont sa popularité et sa disponibilité sur de nombreuses plateformes. Prelude-IDS se restreint sur les plateformes POSIX alors que nous pouvons retrouver Snort aussi sur Windows. A cela s'ajoute l'importance de la base de données des signatures d'attaque de Snort, pour expliquer sa plus grande popularité. Mais Prelude-IDS est de plus en plus connu dans le monde des professionnels de la sécurité.
Une différence importante cependant, au désavantage de Snort : Snort est un NIDS pur alors que Prelude-IDS intègre des fonctionnalités NIDS et HIDS. Ce dernier est un IDS hybride. Dans la suite, nous ne comparerons que ce qui est comparable, c'est-à-dire la partie NIDS de Prelude-IDS et Snort.

- **Moteur d'analyse et banque de signatures**

Les deux font des analyses par recherche de similitudes avec un scénario préalablement définis. Le mode de fonctionnement est alors similaire. Autant pour Prelude-IDS que pour Snort, les solutions sont stables. Prelude-IDS a un soucis de suivi des standards (pour pouvoir utiliser les signatures de d'autres moteurs, échange des messages en XML, IDMEF).
Notons que même en intégrant les alertes de Snort, Prelude-IDS relève une quantité plus importante d'alertes. En regroupant les alertes de même type, nous nous retrouvons alors avec approximativement le même nombre d'alertes (Prelude ayant ses propres règles, il est normal qu'il en trouve un peu plus que Snort). Prelude-NIDS archive aussi les payloads.
Mais malheureusement, la fusion des payloads (ou des alertes) n'est pas plus présente dans Prelude-IDS que dans Snort.Prelude-IDS a l'avantage d'être très modulaire de par son architecture client-serveur. Un manager peut gérer plusieurs sondes, et une sonde peut envoyer ses alertes à plusieurs managers.

- **La remontée d'alertes**

En utilisant ces outils, nous nous rendons compte du fait que Prelude-IDS remonte un nombre plus important d'alertes que Snort. Là où Snort remonte 30 alertes de 2 types différents, Prelude-IDS remonte parfois 2000 alertes de 3 types différents.
Cela est dû au fait que Prelude relève pratiquement chaque trame suspecte, alors que Snort ne va pas relever chaque trame d'une connexion suspecte. De plus, si on combine les signatures d'attaque de Snort avec celles de Prelude, on se rend compte que Prelude à la capacité de détection d'un plus grand nombre d'attaque. Cela est logique étant donné qu'il s'appuit sur les mêmes signatures que Snort en plus des siennes.

- **Infos disponibles dans les alertes**

Dans les deux cas nous avons des remontées d'alertes bien détaillées, qui permettent de connaître la source et la destination d'une attaque, le type d'attaque, un premier classement de sévérité de l'alerte, ou encore un lien vers un bulletin d'alerte officiel.

Dans les deux outils, il est intéressant de constater la présence de niveaux d'alertes permettant d'en déduire du premier coup d'oeil la criticité.
Il est à noter que Prelude-IDS archive aussi les charges utiles des trames suspectes. Cela permettra ensuite dans certains cas à l'administrateur d'analyser le contenu afin d'écarter plus facilement les faux positifs.

- **Intégration d'outils externes**

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 donées. La banque de scénario de Snort peut aussi être récupérée par Prelude-NIDS et ajoutée aux règles de Prelude-IDS. Ainsi, le nombre important de signatures reconnues par Snort (et qui participe à sa popularité) bénéficie à Prelude-IDS.

2. Conclusion

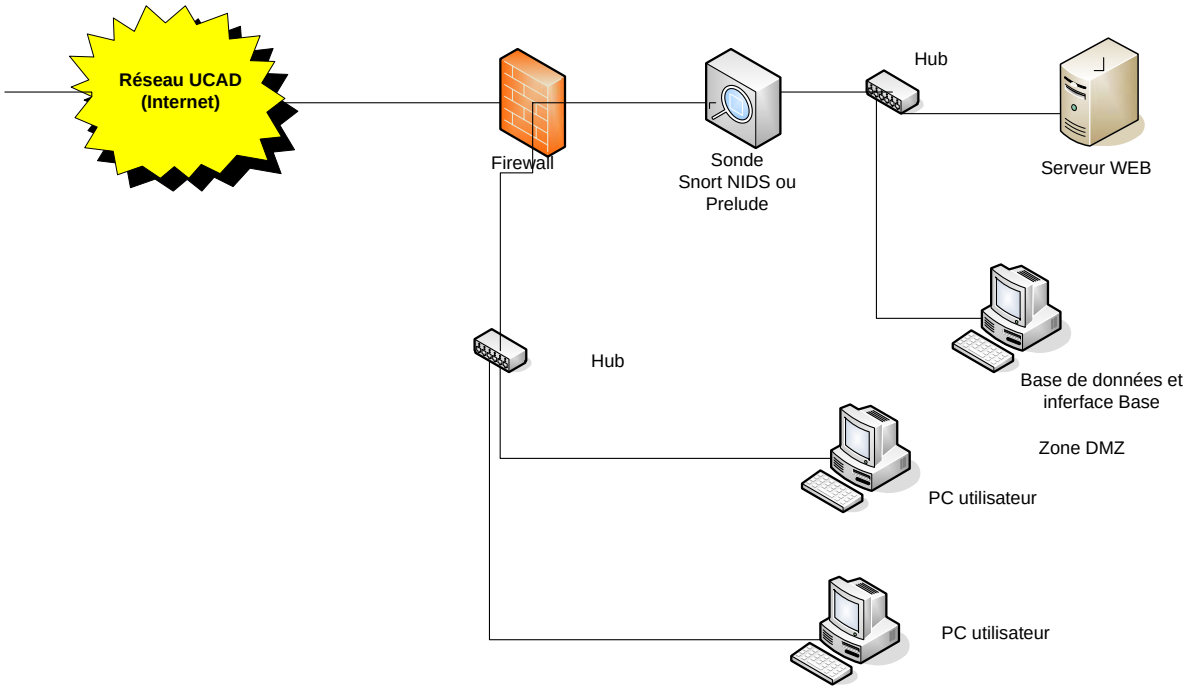
Snort et Prelude sont des outils très fiables et très intéressants dans la mise en place d'une sécurité réseau. Cependant les différences de fonctionnement et de gestion sont nombreuses, ce qui rend la comparaison très difficile. Les avantages de l'un pouvant être considérés comme des inconvénients par l'autre (exemple de la réponse active, ou de l'archivage des payloads). De plus, les règles de Snort pouvant être intégrées dans Prelude-IDS, ce n'est plus sur la quantité d'attaques reconnues que l'on pourra comparer les deux outils. Nous pourrons donc en conclure qu'ils sont très proches et que, suivant l'utilisation que l'on voudra en faire, on choisira soit Snort, soit Prelude-IDS. Notons tout de même que l'interconnexion de Prelude-IDS avec d'autres outils est quand même un avantage considérable sur Snort dans la mise en place d'une solution de sécurité globale mais les avantages de Snort sur Prelude-IDS sont sa popularité et sa disponibilité sur de nombreuses plateformes.
Nous avons donc choisi de concentrer nos efforts sur l'étude et les mises en place de Snort et de Prelude. C'est ainsi que nous avons travaillé sur l'intégration des deux outils dans un réseau de type entreprise (Réseau de test).

Troisième partie

Mise en place de la plate forme de tests

1. Mise en place

1.1 Architecture du réseau de test.



Architecture réseau de test de Snort NIDS

Figure 4

1.2 Mise en place de Snort

1.2.1 Plan d'adressage

Internet	192.168.1.0/24 (En réalité il s'agira des IP publiques fournies par l'UCAD.)
DMZ	192.168.2.0/24
Réseau interne	192.168.3.0/24

1.2.2 Point d'écoutes

Zone DMZ	Une sonde Snort est branchée sur le hub entre les pare feux et la zone DMZ. Cette sonde analyse le trafic en provenance d'Internet vers la DMZ et en provenance du réseau interne vers le DMZ. L'objectif est double : - valider que le filtrage mis en œuvre sur le pare feu A est efficace et protège donc

	les réseaux DMZ et Interne des attaques qui viennent de l'extérieur ; - identifier les attaques ou les anomalies qui viendraient du réseau Interne vers la DMZ ou vers l'extérieur (dans ce dernier cas, on cherchera à détecter les traces liées à des infections virales).
Zone DMZ	Un serveur de base de données est branchée sur le hub ,cette machine héberge la basse données mysql..

L'IDS se compose :
- Une sonde Snort ;
- d'un serveur de base de données qui est utilisé pour stocker les alertes de manière centralisée
- d'une console utilisée pour consulter la base d'alertes et pour administrer la sonde ;

1.2.3 Système d'exploitation

Les systèmes d'exploitation retenue pour la sonde et pour les éléments de l'IDS (SGBD et console) est la distribution Debian avec un noyau 2.4.

Nous avons choisi la distribution Debian pour plusieurs raisons :

- Ses qualités techniques : Debian est réputée pour sa stabilité, pour son très bon système de gestion des dépendances entre les différents composants (ce qui rend l'installation et le retrait des programmes très faciles), et pour sa rapidité à réparer les failles de sécurité ;
- Ses mises a jour faciles.
- Debian GNU/Linux est utilisé par la plupart des fournisseurs d'accès à Internet, comme Free
- Debian est reconnu pour son sérieux et ses fortes prises de positions dans le monde libre. Debian garantit des logiciels open-source !
- Debian est aujourd'hui la seule distribution non-commerciale. Debian est une organisation à but non lucratif constituée d'un millier de développeurs bénévoles répartis sur toute la planète. Elle est dirigée par un project leader élu par les développeurs. Les décisions se prennent au consensus ou par vote.

1.2.4 Méthode d'installation des logiciels

Pour installer les logiciels utilisés pour construire l'IDS nous utilisons les systèmes de paquetages propres à la distribution Debian,les paquetages seront installés à l'aide de la commande apt-get.

1.2.5. Les sondes

La sonde est constituée d'une machine équipée de deux cartes réseau :
- une interface activée en mode "stealth", c'est-à-dire sans adresse IP ;
- une interface activée sur un réseau dédié à l'IDS. Ce réseau permet la remontée des alertes vers une base de données unique ainsi que l'administration des sondes.
Le logiciel utilisé est Snort en version 2.6.0.2 téléchargeable sur le site www.snort.org

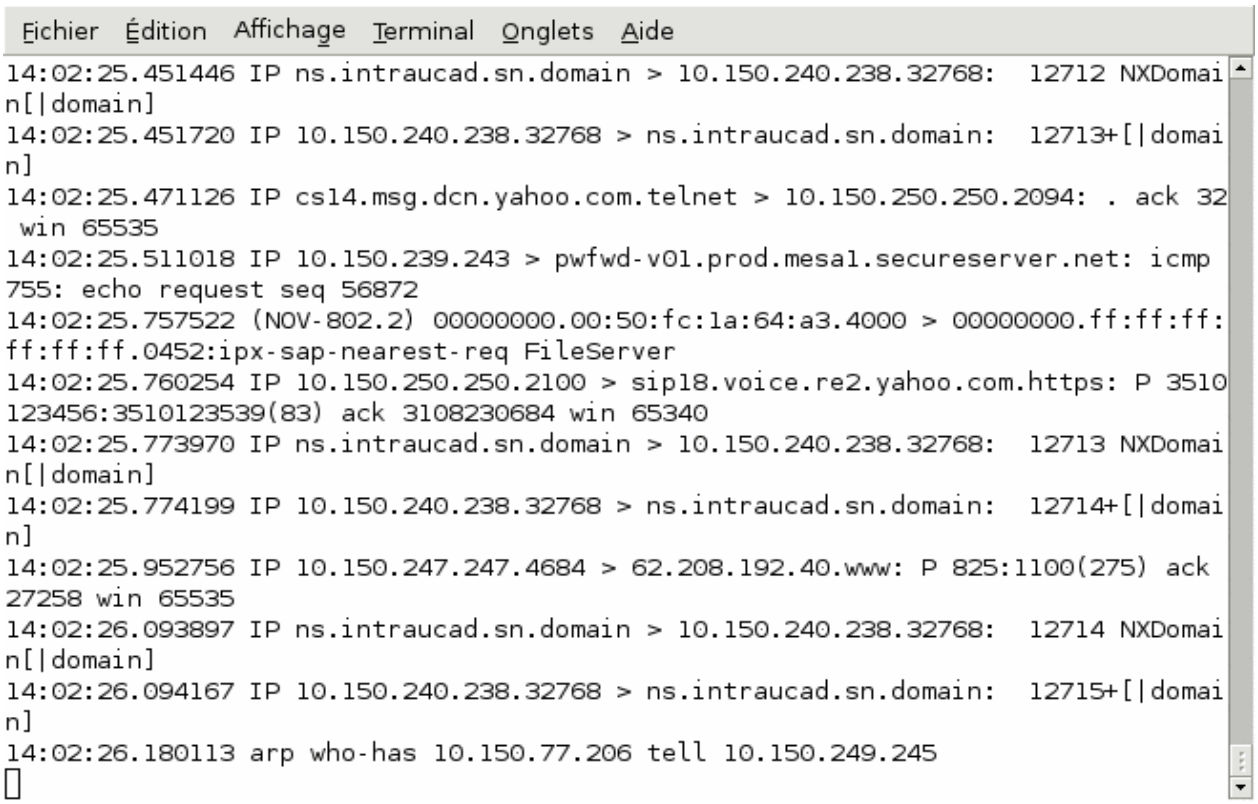
- **Interface en mode stealth**

Une interface en mode stealth est une interface réseau active mais à laquelle aucune adresse IP n'a été attribuée.
Le trafic réseau peut être lu depuis cette interface mais elle ne peut pas en générer.
Sous Linux, pour activer une interface sans lui attribuer d'adresse IP, il faut utiliser la commande ifconfig de la manière suivante :

 # ifconfig eth1 up

L'interface eth1 est alors montée mais sans adresse IP :
Pour vérifier que le trafic est bien visible depuis cette interface, vous pouvez utiliser le logiciel tcpdump

 # su –c tcpdump



- **Sécurité**

La sécurité de la sonde s'appuie sur :
- une désactivation du routage IP au niveau du noyau ; dans le fichier /etc/sysctl.conf, la valeur du paramètre net.ipv4.ip_forward doit être 0. Si nécessaire, il est toujours possible de recompiler le noyau Linux en retirant la fonctionnalité ip_forwarding.

- Une règle NetFilter/IPTables qui interdit toute émission de paquet sur l'interface activée en mode stealth (iptables -A OUTPUT -o eth1 -j DROP).
Ces deux principes rendent la sonde invisible depuis les points d'écoute. Cela ne signifie cependant pas qu'elle est à l'abri d'une attaque en aveugle (cas où un attaquant, soupçonnant la présence d'une sonde, envoie du trafic volontairement malformé dans le but de leurrer la sonde ou de la rendre inopérante..

1.2.6 Le réseau dédié à l'IDS

Le réseau IDS est utilisé pour administrer la sonde , remonter les alertes depuis la sonde vers le serveur SGBD et mettre à jour les règles.
Il doit être sécurisé de manière à ne pas compromettre la sécurité du S.I. en faisant office de passerelle à un éventuel attaquant. Dans le cas présent, nous avons choisi de dédier un réseau non interconnecté aux autres zones du S.I.. Il est cependant possible de relier le réseau IDS à une autre zone du S.I. en passant par un pare feu. La sécurité de cette interconnexion doit alors être soigneusement validée et surveillée.

1.2.7 Serveur SGBD

Il est judicieux de centraliser les alertes sur une seule machine surtout dans le cas ou on utilise plusieurs sondes. De même, l'utilisation d'une base de données présente de nombreux avantages et offre la possibilité de consulter les alertes depuis une interface dynamique et conviviale.
Dans notre cas, nous utilisons un serveur de type NEC (ordinateur P4 ,disque dure 40 G ,RAM de 512M)sous Linux et un serveur de bases de données MySQL.

1.2.8 Console

La console est elle aussi hébergé sur la machine ou se trouve la base de données. Un serveur HTTP Apache avec l'interpréteur PHP est utilisé pour consulter le contenu de la base de données. On utilise l'interface BASE pour cela.

1.2.9 Administration

Sur chaque machine qui compose l'IDS, un démon SSH OpenSSH seras utilisé pour les tâches d'administration courantes.

1.2.10 Installation du serveur SGBD

La première étape de la construction de notre IDS va consister à installer le serveur SGBD qui sera utilisé pour stocker les alertes.
Nous allons utiliser une base de données de type MySQL pour stocker les alertes générées par la sonde. Le logiciel retenu est MySQL dans sa version 4.1
Pour installer MySQL sous Debian, nous allons utiliser la commande

apt-get :
**apt-get install mysql-server-4.1 mysql-common **
mysql-client-4.1

1.2.11 Sécurisation

Par défaut, certains paramètres de MySQL sont inutiles ou trop permissifs. Il est donc nécessaire de les modifier.
Les actions suivantes doivent être entreprises avant la mise en production effective du serveur :
- Destruction des comptes et bases de test ;
- Renforcement du contrôle d'accès au compte root : cet utilisateur ne doit pouvoir se connecter à la base que depuis la machine en locale (IP source : 127.0.0.1). Pour les opérations d'administration à distance, on créera un autre compte (par exemple : myadmin) dont les droits seront plus élevés que ceux d'un utilisateur non privilégié mais plus restreints que ceux de root. Cet utilisateur pourra créer des bases, les administrer, mais ne pourra pas créer de nouveaux utilisateurs ni modifier les droits des utilisateurs existants ni intervenir sur les processus MySQL.

- **Configuration du serveur MySQL**

MySQL utilise un système de gestion de comptes Utilisateurs pour définir les droits attribués sur les objets MySQL (c'est-à-dire : les bases de données, les tables dans les bases) et les commandes associées (création de base, de tables, utilisation des ordres SQL INSERT, UPDATE, DELETE, etc.).
Comme sous Unix, le super utilisateur MySQL s'appelle root et par défaut ce compte n'est pas sécurisé : il n'a en effet pas de mot de passe.
La première étape de la configuration du serveur MySQL va donc consister à attribuer un mot de passe à cet utilisateur root :
- Vérifier que le démon mysqld est bien lancé ou bien le démarrer ;
- Lancer la commande suivante depuis la ligne de commande :

 # mysqladmin -u root password "mypassword"

mypassword doit bien entendu être remplacé par un mot de passe sûr et gardé secret.

- Vérifier qu'il est maintenant nécessaire d'utiliser ce mot de passe pour les connexions MySQL :

 # mysql –u root –p
Enter password : ****
mysql>

- Créer la base snort et les utilisateurs autorisés à s'y connecter :

 # mysql> CREATE DATABASE snort ;
 # mysql> GRANT CREATE, INSERT, SELECT, DELETE, UPDATE
ON snortdb.* TO snorty IDENTIFIED BY 'snorty' ;
 # mysql> FLUSH PRIVILEGES ;

Le schéma de la base snort est contenu dans le fichier
create_mysql. Qui se trouve dans le répertoire décompresser nommé snort-2.6.0.2/cshema Il faut charger ce schéma dans la base snort pour la rendre opérationnelle :
Il faut se ^lacer dans le répertoire /usr/local/snort-2.6.0.2/cshema avant de taper la commande suivante .

 # mysql –u snorty –p snortdb < create_mysql
Enter password: ****
Pour vérifier que le schéma est bien présent :
 # mysql –u snorty –p snortdb
Enter password: ****
mysql> show tables;

Nous pouvons maintenant passer à l'étape suivante : installer la sonde.

1.2.12 installations de la snort

Ce paragraphe décrit l'installation d'une sonde Snort.

- **Installation de Snort 2.6.0.2**

Le logiciel Snort doit être installé la sonde. La version retenue est la 2.6.0.2 dernière version stable disponible en Août 2006.
Ce logiciel s'appuie sur les fonctionnalités de la bibliothèque libpcap pour capturer les paquets. Il faut donc installer cette bibliothèque avant d'installer Snort. La version de la libpcap à jour en avril 2006 est la 0.9.2.
De la même façon, Snort s'appuie sur les fonctionnalités de la bibliothèque PCRE (Perl Compatible Regular Expressions) pour rechercher des chaînes de caractères dans les paquets. La dernière version à jour en avril 2006
est la 6.4.
Pour installer snort 2.6.0.2 sous Debian, nous allons utiliser la commande apt-get
Pour installer les deux bibliothèques Libpcap et Libpcre et leurs dépendances :

**# apt-get install libnet1 libnet1-dev libpcre3 **

**libpcre3-dev autoconf automake1.9 **
libpcap0.8 libpcap0.8-dev libmysqlclient14-dev gcc make

Si cette commande ne marche pas il va falloir télécharger le deux librairies puis les installés.

1ère étape : Compilation

Note : l'option –with-mysql est requise pour que la binaire snort puisse utiliser une base de données MySQL pour stocker les alertes.

 # cd /usr/local/src
 # tar xvzf snort-2.6.0.2.tar.gz
 # cd snort-2.6.0.2
 # ./configure --with-mysql
 # make
 # make install

2ème étape : Création des répertoires et de l'environnement de travail
 # mkdir /etc/snort
 # mkdir /var/log/snort
 # groupadd snort
 # useradd -g snort snort
 # chown snort:snort /var/log/snort
 # cd /etc/snort
 # mv /root/snortrules-2.6.0.2..tar.gz.sig .
 # tar xvzf snortrules2.6.0.2
 # cp /usr/local/src/snort-2.6.0.2/etc/*.conf* .
 # cp /usr/local/src/snort-2.6.0.2./etc/*.map .

- **Validation**

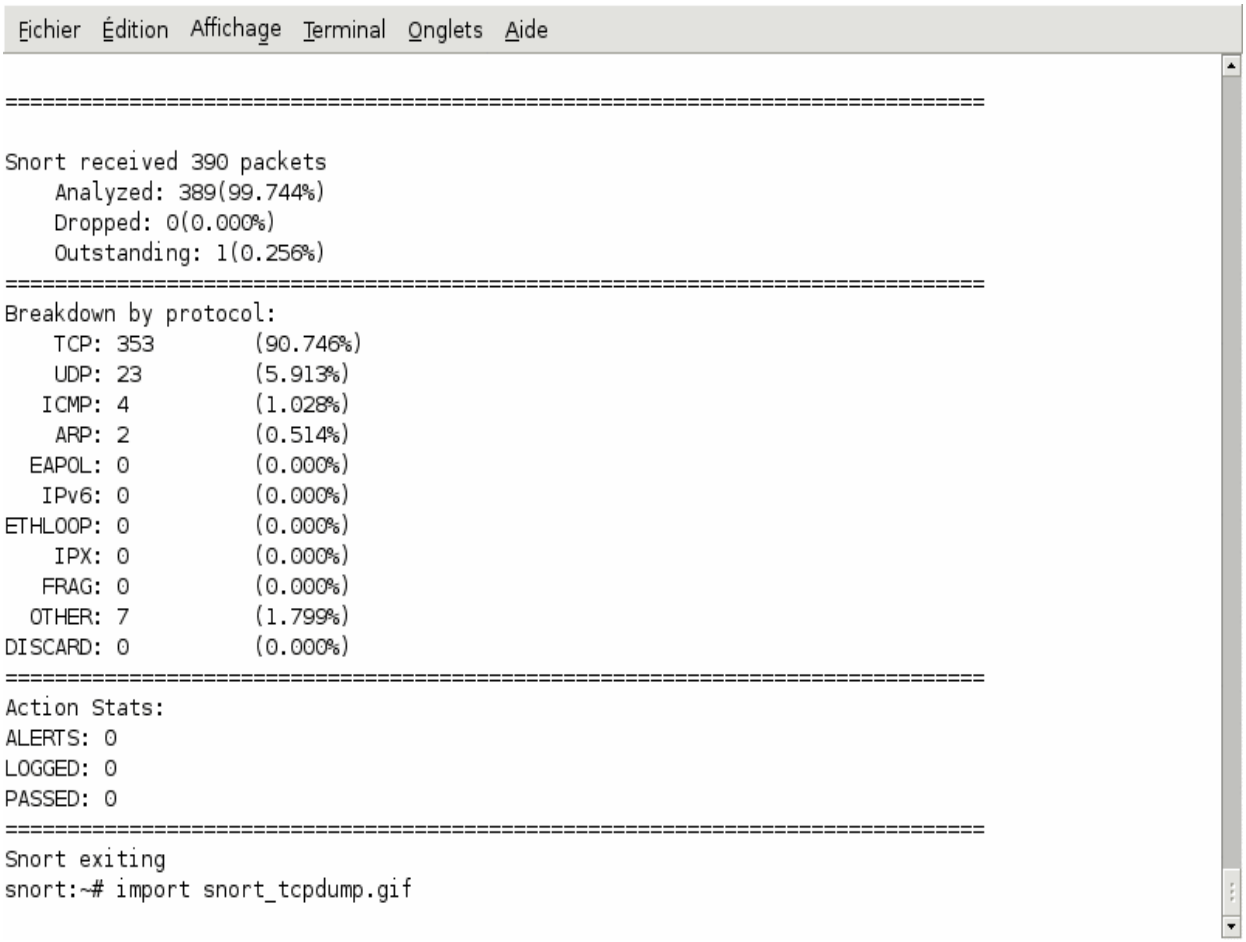
Pour vérifier que le binaire snort fonctionne, lancez la commande suivante :
 # snort -V
La sortie est la suivante :

- **Utilisation de Snort comme Analyseur de trafic**

Snort peut être utilisé pour lire le trafic réseau sur une interface de la manière suivante :

 # snort –dev –i eth0

Cette commande et ces options affichent à l'écran le trafic qui transite sur le réseau auquel est connecté l'interface eth0 :



Il est également possible de capturer le trafic réseau et de le stocker sur disque dans un fichier au format pcap :

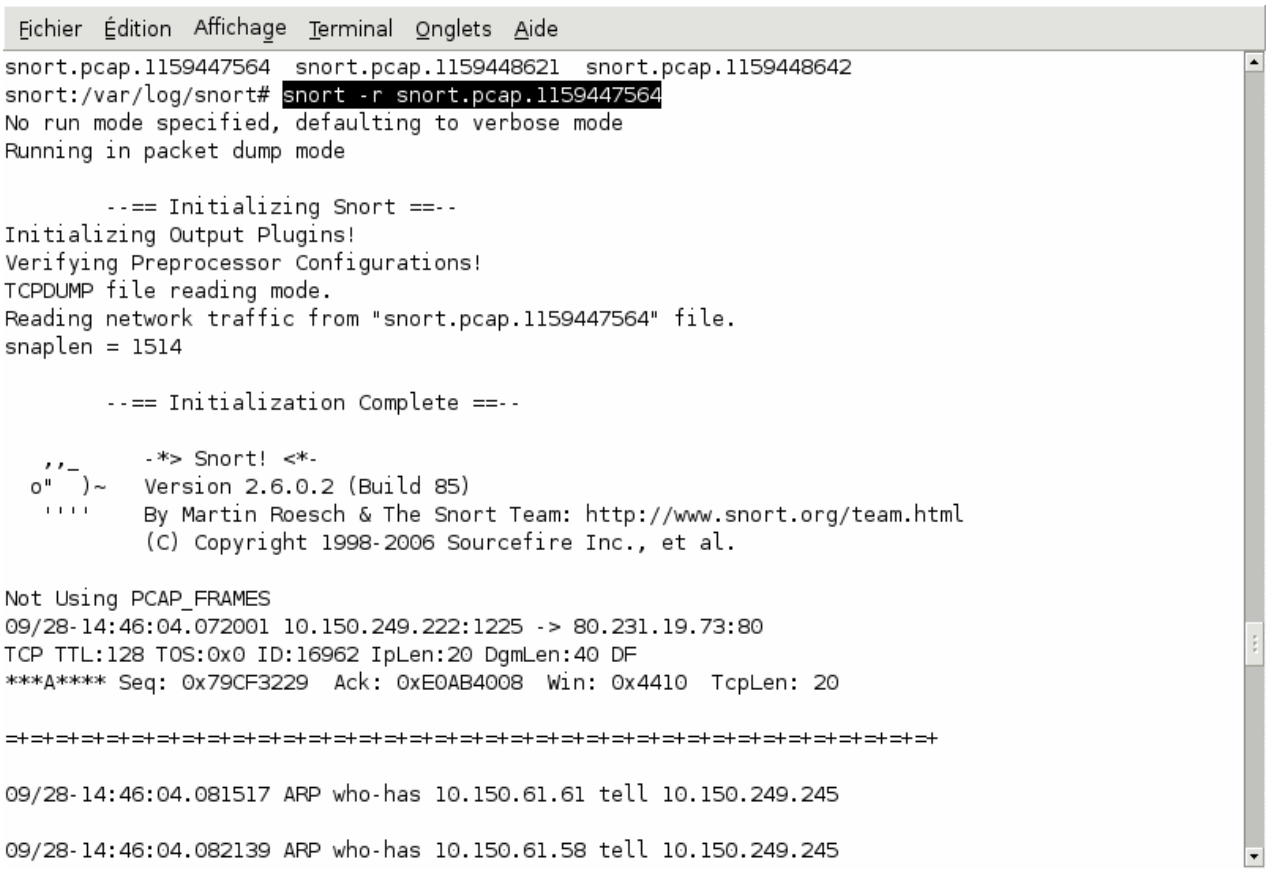
 # snort -b -i eth0 -L snort.pcap

Le trafic est stocké dans le fichier préfixé snort.pcap et suffixé par un “timestamp” Unix dans le répertoire /var/log/snort :

 # ls /var/log/snort/snort.pcap.1147077043

Snort peut alors être utilisé pour lire ce fichier et en afficher le contenu :

 # snort –dev –r /var/log/snort/ snort.pcap.1159447564



- **Exemple de scan de port avec nmap**

Aujourd'hui, les scans sont devenus des attaques courantes, Les scans servent principalement à obtenir de l'information sur un hôte, un réseau (pour préparer une attaque plus élaborée).
Les informations qu'un attaquant peut obtenir sur votre réseau figurent dans le résultat du scan ci-dessous (nmap - un simple exemple qui ne montre pas toutes les possibilités de nmap)
Principalement, vous obtenez ce qui suit :

- le système d'exploitation de la machine
- les versions de certains programmes actifs
- les services actifs qu'il est possible d'attaquer
- les ports ouverts

La sortie suivante de nmap est celle de l'un de mes hôtes, immédiatement après une installation par défaut. Je n'ai volontairement fait aucun changement (dans la configuration) :

 # nmap -sT localhost

Starting nmap V. 2.54BETA36 (www.insecure.org/nmap/)
Interesting ports on snort (127.0.0.1):

(The 1552 ports scanned but not shown below are in state: closed)

Port	State	Service
21/tcp	open	ftp
23/tcp	open	telnet
80/tcp	open	http
111/tcp	open	sunrpc
113/tcp	open	auth
6000/tcp	open	X11

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second

- **Utilisation de Snort comme NIDS**

C'est cette fonctionnalité de Snort qui nous intéresse dans notre cas d'études

Nous voulons utiliser Snort comme sonde de détection Réseau.
Pour cela, il nous faut :

- éditer et modifier les paramètres du fichier de configuration snort.conf ;
- charger les règles Snort à jour et choisir les types de règles à utiliser ;
- lancer Snort en mode démon.

- **Configuration**

Le fichier de configuration de Snort se trouve dans /etc/snort sous Debian. Il s'agit d'un simple fichier texte qui obéit aux règles standard des fichiers de configuration du monde Unix :

- toute ligne qui commence par le caractère # est un commentaire
- les variables sont appelées en faisant précéder leur nom du caractère $
- d'une manière générale les paramètres sont définis sur une seule ligne et il est possible, pour une meilleure lisibilité de passer à la ligne en insérant le caractère \.

Dans le fichier snort.conf, voici les paramètres les plus significatifs :
HOME_NET : ce paramètre désigne les adresses IP internes de nos réseaux, c'est-à-dire les adresses cibles des attaques.
Dans notre cas, ce paramètre prendra les valeurs suivantes :

EXTERNAL_NET : il s'agit des adresses des réseaux dits « externes », c'est-à-dire les adresses des sources des attaques d'une manière générale.
Une pratique courante consiste à déclarer ce paramètre comme étant l'inverse du paramètre HOME_NET :

var EXTERNAL_NET !$HOME_NET
Dans notre cas, il prendra les valeurs suivantes :

Les autres paramètres importants concernent le mode de stockage des alertes et les règles chargées au lancement du démon snort.

Nous utilisons une sonde mais nous stockerons les alertes sur une base de données . Cette base est de type MySQL.
Par précaution, il est recommandé d'utiliser, en plus de cette base de données, les mode de stockage suivants :

- **tcpdump :**

chaque paquet qui génère une alerte sera conservé sur disque sur chaque sonde. En effet, l'interface BASE que nous allons utiliser pour visualiser le contenu de la base de données MySQL ne permet pas d'afficher toutes les informations contenues dans les paquets capturés. Notamment les
informations de niveau 2 (couche Ethernet) ne sont pas lisibles. Or dans certains cas (attaques depuis le même segment réseau avec usurpation de l'adresse IP de l'attaquant) les informations relatives à cette couche peuvent être très utiles : c'est le cas par exemple des adresses MAC des machines sources.

- **CSV :**

ce mode stocke les alertes dans un fichier texte au format CSV (chaque alerte est stockée sur une ligne, chaque donnée qui compose l'alerte est séparée des autres par une virgule). Ce format permet l'injection facile dans une base de données. Il servira de « backup » en cas de problème sur la base de
données (perte de la liaison entre la sonde et le serveur SGBD, corruption de la base, etc.).

- **Syslog :**

chaque alerte donnera lieu à l'insertion d'une ligne dans les journaux du système.
Les paramètres suivants sont utilisés pour cela :

Sylog :

output alert_syslog: LOG_AUTH LOG_ALERT

TcpDump :

output log_tcpdump: tcpdump.log

CSV :

output alert_csv : /var/log/snort/alert.csv Base MySQL :
**output database: log, mysql, user=snorty password=test **
dbname=snortdb host=adresse_ip sensor_name=nom_de_la_sonde

Notes :

L’option sensor_name est importante. C’est cette valeur qui sera utilisée dans l’interface BASE pour indiquer la sonde sur laquelle une alerte a été générée.
La valeur des paramètres user, password, dbname et host sera identique sur chaque sonde. Dbname est le nom de la base MySQL utilisée pour stocker les alertes, Host l’adresse IP sur serveur SGBD.

Les signatures

Snort utilise des signatures pour détecter les tentatives d’intrusion ou les anomalies dans le trafic réseau.
Ces signatures sont fournies par le projet Snort et le projet Bleeding-Edge.
Elles se présentent sous la forme de fichiers texte qui contiennent une signature par lignes. Elles sont généralement regroupées par types et sont stockées sur les sondes dans le répertoire /etc/snort/rules.
Dans le fichier de configuration snort.conf, les signatures sont activées (chargées au lancement du démon snort) en décommentant la ligne qui indique quels sont les fichiers de signatures à utiliser :

 # include $RULE_PATH/info.rules
 # include $RULE_PATH/icmp-info.rules
include $RULE_PATH/virus.rules

Dans l’exemple ci-dessus, les signatures du fichier /etc/snort/rules/virus.rules sont chargées, mais pas celles des
fichiers info.rules ni icmp-info.rules.
La variable RULE_PATH désigne le répertoire où sont stockés les fichiers de signatures :

var RULE_PATH /etc/snort/rules

- **Lancement du démon Snort**

Une fois le fichier de configuration snort.conf modifié selon les besoins, le démon Snort est lancé à l’aide du **script /etc/init.d/snortd ou par la commande**

 # /usr/local/bin/snort -Dq -u snort -g snort -c /etc/snort/snort.conf

1.2.14 installations de la console

La console est le dernier élément qu’il nous faut installer.
Elle s’appuie sur un serveur Apache et l’interpréteur PHP. Ces deux logiciels nous permettront d’installer l’interface BASE pour visualiser le contenu de la base de données dans laquelle sont stockées les alertes.

- **Installation du serveur Apache**

Nous allons utiliser un serveur Apache en version 2 ainsi que l'extension SSL qui nous permettra de sécuriser les consultations.
Pour installer Apache sous Debian, nous allons utiliser la commande apt-get :
apt-get install apache-ssl apache-common libssl-dev.

- **Installation de l'interpréteur PHP**

Pour installer PHP sous Debian, nous allons utiliser la commande apt-get :

**apt-get install libapache-mod-php4 php4-mysql php4-gd **
php4-pear libphp-adodb.

- **Installation de l'interface BASE**

BASE est une interface pour Snort écrite en PHP. La dernière version à jour .
Note : dans la suite de ce paragraphe, on partira du principe que le DocumentRoot du serveur Apache est le répertoire /var/www.

- **Pré requis**

BASE utilise deux bibliothèques : AdoDB pour ses fonctions de connexion et d'interrogation à des bases de données de plusieurs types, dont MySQL, et GD pour ses fonctions graphiques.
L'installation de ces deux bibliothèques peut s'effectuer à partir des paquetages ou du code source, sachant que ce dernier est une simple archive Tar qu'il faut décompresser dans le DocumentRoot du serveur Apache.

- **Installation de BASE**

L'interface BASE est fournie elle-aussi sous forme d'une archive Tar. L'installation de BASE consiste à décompresser cette archive et l'extraire dans le DocumentRoot du serveur Apache.
Une fois extraite, il est recommandé de créer un lien symbolique qui pointe sur le répertoire base-1.2.4 :

 # ln –s base-1.2.4 base

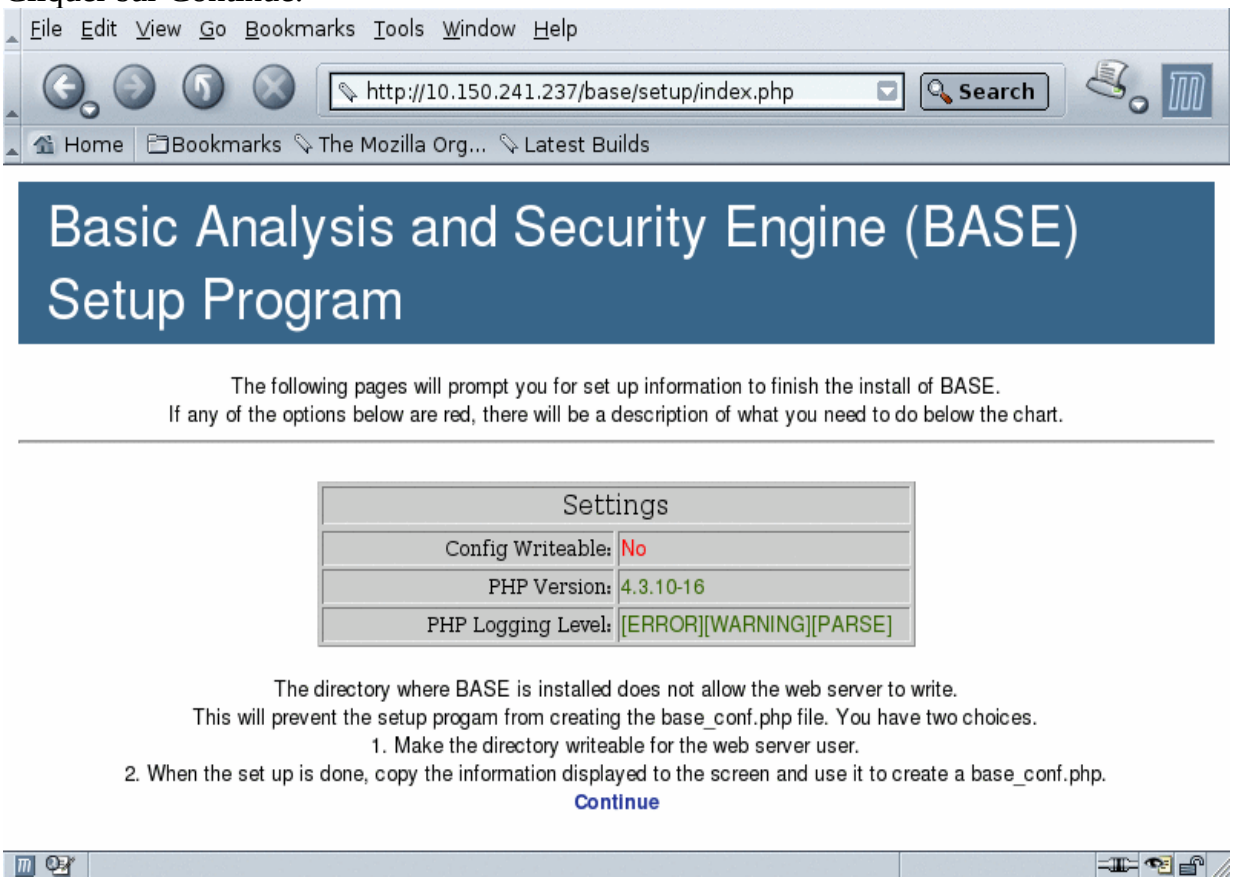
- **Configuration de BASE**

La configuration de l'interface BASE se fait depuis un navigateur :

- vérifier que le serveur Apache est bien lancé sur le serveur où BASE a été installée, le démarrer le cas échéant ;
- depuis un navigateur Web, ouvrir l'URL http://adresse_du_serveur/base

Il suffit alors de suivre les écrans et les instructions qu'ils contiennent :

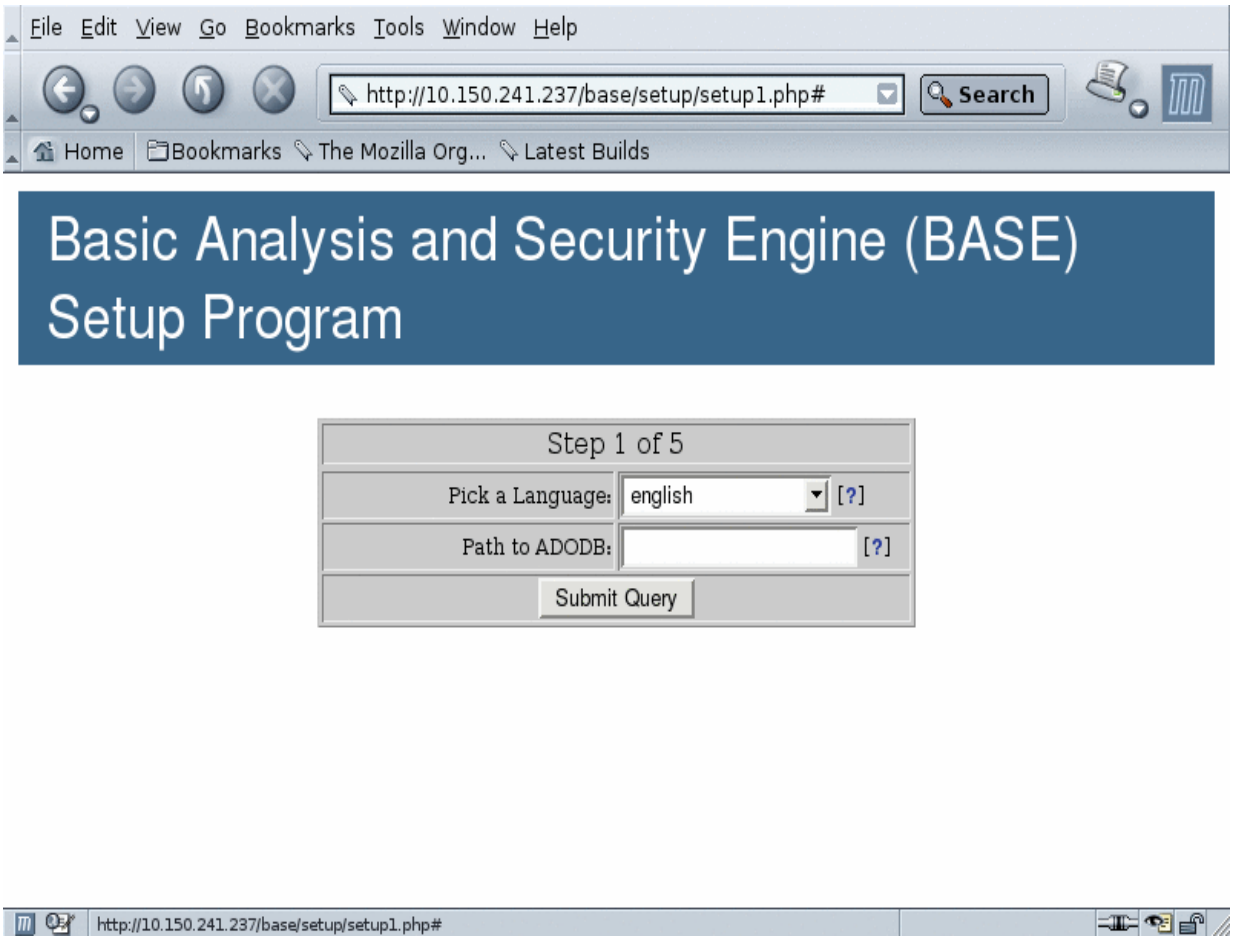
Cliquer sur Continue.



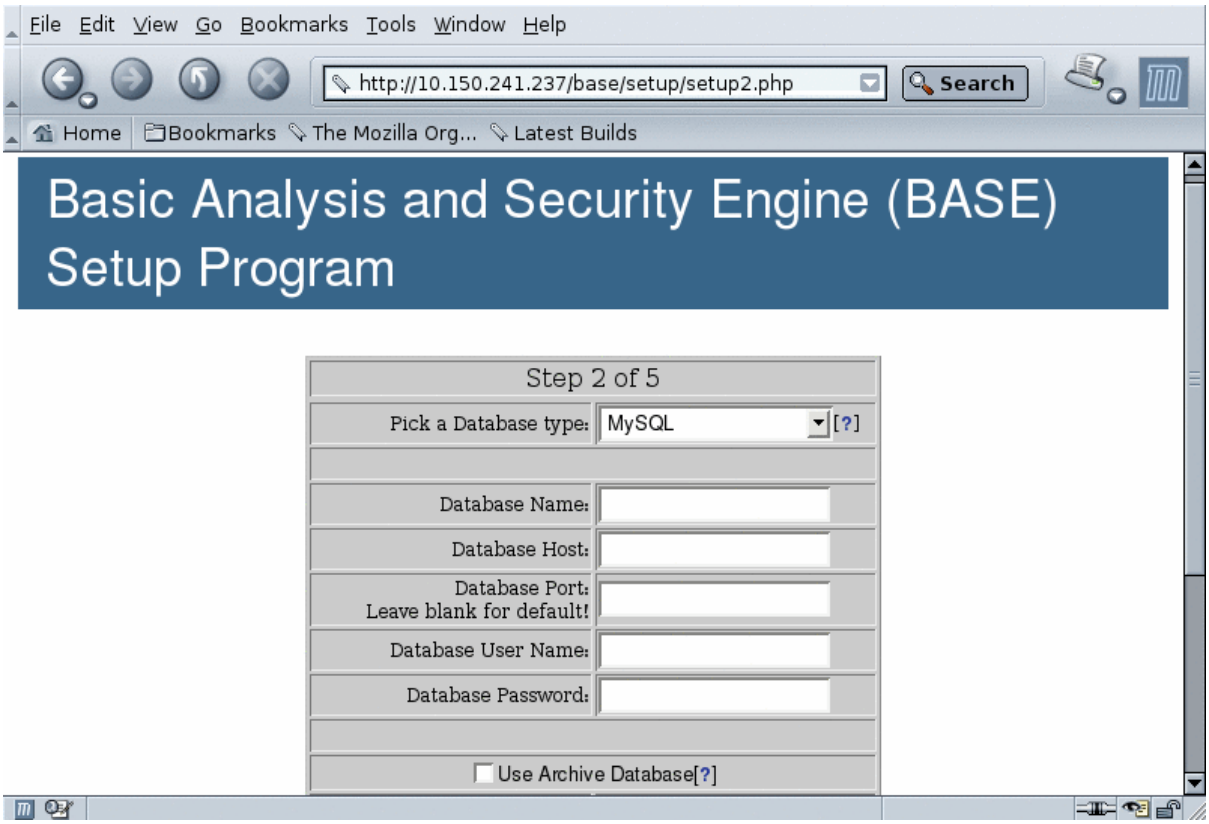
Note : Si le champ Config Writeable est à No, il faut modifier les attributs des fichiers du répertoire
/var/www/base :

$ chown –R apache:apache /var/www/base

Choisir à l'étape suivante la langue retenue et renseigner le chemin d'accès aux librairies PHP AdoDB :



A l'étape suivante, renseigner le nom de la base de données utilisée pour stocker les alertes générées par les sondes (Champ Database Name), le nom ou l'adresse IP du serveur qui héberge le serveur MySQL (Champ Database Host), le nom d'utilisateur MySQL et le mot de passe pour accéder à la base (Champs Database User Name et Database Password) :


File Edit View Go Bookmarks Tools Window Help
http://10.150.241.237/base/setup/setup2.php
Search
Home
Bookmarks
The Mozilla Org...
Latest Builds
Basic Analysis and Security Engine (BASE)
Setup Program
Step 2 of 5
Pick a Database type:
MySQL
[?]
Database Name:
Database Host:
Database Port:
Leave blank for default!
Database User Name:
Database Password:
Use Archive Database[?]

1.2 Démarche de mise en place de Prelude NIDS

Nous expliquons l'installation et la configuration de Prelude-IDS et de tous les paquetages qu'il utilise.
La plateforme de travail est Debian et nous proposons d'utilisons les versions suivantes des sources de Prelude-IDS :
libprelude-0.8.4, prelude-manager-0.8.6, prelude-nids-0.8.1, et prelude-lml-0.8.2 .
Nous choisirons de stocker les alertes reçues par le manager dans une base de donnée MySQL.
En exsenple nous utilisons une machine d'adresse IP 192.168.0.2

1.2.1 Paquetages nécessaires

Il est nécessaire de télécharger, sur le site **www.prelude-ids.org** les paquetages suivants :

libprelude-x.x.x.tar.gz (nous utiliserons ici la version 0.8.4) ;
prelude-manager-x.x.x.tar.gz (nous utiliserons ici la version 0.8.6) ;
Prelude-nids-x.x.x.tar.gz (nous utiliserons ici la version 0.8.1) ;
prelude-lml-x.x.x.tar.gz (nous utiliseront ici la version 0.8.2) .

Libprelude est nécessaire à **prelude-nids, prelude-lml, et prelude-manager**.

Décompressez les fichiers dans **/usr/local/src/**.
Nous avons maintenant le nouveau répertoire

libprelude-0.8.4, prelude-lml-0.8.2, prelude-manager-0.8.6, et prelude-nids-0.8.1 .

1.2.2 Installation de paquetages au préalable

Prelude-IDS nécessite pour fonctionner, des bibliothèques et autres applications que nous allons installer maintenant.

1.2.3 Paquetages nécessaires à Prelude-IDS

– gtk-doc-tools nécessaire pour libprelude, donc aussi à prelude-nids, prelude-lml, et prelude-manager
– **libssl-dev** nécessaire pour libprelude, donc aussi à prelude-nids, prelude-lml, et prelude-manager

– **mysql-server** à installer sur le poste hébergeant le manager, si l'on veut stocker les alertes dans une bases MySQL

– **libmysqlclient10-dev** nécessaire à prelude-manager
– **libxml2-dev** nécessaire à prelude-manager

– **libpcre3-dev** nécessaire pour prelude-lml
– **libfam-dev** nécessaire pour prelude-lml

1.2.4 Installation de ces paquetages

apt-get install gtk-doc-tools
apt-get install libssl-dev
apt-get install mysql-server

Pendant l'installation, il est proposé de démarrer le serveur MySQL au démarrage de la machine. Il est conseillé de répondre **Yes**.

apt-get install libmysqlclient10-dev
apt-get install libxml2-dev
apt-get install libpcre3-dev
apt-get install libfam-dev

1.2.5 Installation de libprelude

Rentrez dans le répertoire **/usr/local/src/libprelude-0.8.4/** et tapez les commandes suivantes :

./configure --enable-gtk-doc --enable-openssl
make
make install

1.2.6 Installation du manager

Rentrez dans le répertoire /usr/local/src/prelude-manager-0.8.6/ et tapez les commandes suivantes :

./configure --enable-gtk-doc --enable-mysql --enable-openssl

1.2.7 Installation de la sonde réseau (nids)

Rentrez dans le répertoire /usr/local/src/prelude-nids-0.8.1/ et tapez les commandes suivantes :

./configure --enable-gtk-doc
make
make install

1.2.8 Installation de la sonde hôte (lml)

Rentrez dans le répertoire **/usr/local/src/prelude-lml-0.8.2/** et tapez les commandes suivantes :

./configure --enable-gtk-doc --enable-fam
make
make install

1.2.9 Configuration

Nous allons aborder ici la configuration de tous les éléments constituant Prelude-IDS. Certaines valeurs comme les adresses IP ou encore les mots-de-passe, seront à adapter par chacun.

1.2.10 Configuration de MySQL

Lancer le démon mysql si cela n'a pas encore été fait :

/etc/init.d/mysql restart

Rentrez dans mysql :

mysql

Créez la base de données prelude

create database prelude ;

puis ressortir :

quit

Donner les droits en exécution (550 par exemple) au fichier

/usr/local/src/prelude-manager-0.8.6/prelude-manager-db-create.sh

Exécutez maintenant le script :

/usr/local/src/prelude-manager/prelude-manager-db-create.sh

Il y a 6 phases (de 0 à 5).
La première, répondre **y.**
À la deuxième, répondre **mysql**.
À la troisième,répondre **prelude** .
Pour la quatrième phase qui est la numéro **3**,
indiquez root (choix par défaut) pour l'administrateur de la base, et laisser le **mot-de-passe** vide car par defaut l'utilisateur root n'a pas de mot-de-passe par défaut (cela sera à changer ultérieurement pour améliorer la sécurité du système). L'avant-dernière

phase attend la réponse **prelude** (choix par défaut),. Enfin, pour la dernière phase, répondre **yes** si toutes les informations saisies sont correctes.

1.2.11 Configuration du manager

Éditez le fichier de configuration du manager
/usr/local/etc/prelude-manager/prelude-manager.conf .

Ensuite, faire apparaître les lignes suivantes (les décommenter si elles sont en commentaires, sinon, les écrire dans le fichier) :
Dans la rubrique Prelude Manager :

 # Indiquer ici l'adresse et le port d'écoute des sondes par le serveur.
#Par défaut, c'est le port 5554 qui est utilisé.
sensors-srvr = 192.168.0.2 ;
Dans la rubrique MySQL :
 # La ligne ci-dessous est valable même si le manager
#et les sondes ne sont pas sur la même machine.
dbhost = localhost ;
dbname = prelude ;
dbuser = prelude ;
dbpass = dessstri ;

1.2.12 Configuration de la sonde réseau (nids)

Éditez le fichier de configuration de la sonde réseau
/usr/local/etc/prelude-nids/prelude-nids.conf .

Ensuite, faire apparaître les lignessuivantes (les décommenter si elles sont en commentaires, sinon, les écrire dans le fichier) :

manager-addr = 192.168.0.2 ;
user = prelude ;

Ici, le manager se trouve sur la même machine, sinon, indiquez l'adresse IP de la machine hébergeant le manager. Au cas où le port de communication par défaut serait changé, il faudra l'indiquer à la suite de l'adresse IP, en séparant l'adresse IP et le port de communication de 2 points :

manager-addr = 192.168.0.2 :5554 ;

1.2.13 Configuration de la sonde hôte (lml)

Éditez le fichier de configuration de la sonde hôte

/usr/local/etc/prelude-lml/prelude-lml.conf .

Ensuite, faire apparaître les lignes suivantes (les décommenter si elles sont en commentaires, sinon, les écrire dans le fichier) :

manager-addr = 192.168.0.2 ;

Ici, le manager se trouve sur la même machine, sinon, indiquez l'adresse IP de la machine hébergeant le manager. Au cas où le port de communication par défaut serait changé, il faudra l'indiquer à la suite de l'adresse IP, en séparant l'adresse IP et le port de communication de 2 points :

manager-addr = 192.168.0.2 :5554 ;

1.2.14 Lancement de l'écoute

Sur le manager, tapez la commande suivante, dans un terminal, pour créer un utilisateur et avoir un

mot-de-passe :
manager-adduser

Il sera donné un mot-de-passe.
Sur la sonde, tapez la commande suivant, dans un terminal, pour lancer l'ajout d'un utilisateur dans le sensor :
Pour la sonde réseau, avec 192.168.0.2 pour adresse du manager :

sensor-adduser -s prelude-nids -m 192.168.0.2 -u 0

Pour la sonde hôte, avec 192.168.0.2 pour adresse du manager :

sensor-adduser -s prelude-lml -m 192.168.0.2 -u 0

Dans un cas comme dans l'autre, la suite est la même. Il est demandé de rentrer le mot-de-passe noté (lors du manager-adduser), puis d'indiquer le nom de l'utilisateur (prelude) et son mot-de-passe (dessstri). Ensuite on accepte de créer cet utilisateur.

On lance ensuite le manager par la commande :

prelude-manager

Nous pouvons nous affranchir du fichier de configuration en donnant les paramètres en arguments. Dans
notre cas, cela va donner ceci :

prelude-manager --mysql --dbhost localhost --dbname prelude --dbuser prelude --dbpass dessstri

Puis on lance la sonde avec la commande pour la sonde réseau (eth0 est l'interface d'écoute du réseau) :

prelude-nids -i eth0 -u root

ou avec la commande suivante pour la sonde hôte :

prelude-lml -u root

Il est à noter qu'il faut créer un nouvel utilisateur (manager-adduser) pour chaque nouvelle sonde. En revanche, un manager peut écouter, en même temps, les remontées d'alertes de plusieurs sondes.

1.2.15 Installation et configuration du prelude-php-frontend

Étant donné que le prelude-php-frontend se base sur un serveur web, nous installerons **Apache-ssl,** ainsi que **php4.**

apt-get install apache-ssl php4 php4-mysql

Accepter l'ajout de extension=**mysql.so** qui est proposé pendant l'installation.
Ensuite se placer dans le répertoire contenant les sources compressées du prelude-php-frontend, les décompresser, et les mettre dans le répertoire du serveur Apache-ssl :

cd /usr/local/src/

tar -xzf prelude-php-frontend-0.8.1.tar.gz

Est maintenant apparu le répertoire prelude-php-frontend que nous copions dans le répertoire du serveur web (/var/www/par défaut).

cp -r prelude-php-frontend /var/www/

1.2.16 Configuration

Éditez le fichier de configuration d'Apache **(/etc/apache-ssl/httpd.conf**), et y écrire où décommenter les lignes suivantes (la machine a pour adresse IP 192.168.0.2):

Listen 192.168.0.2
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
DirectoryIndex index.php index.html index.htm
ServerName localhost
DocumentRoot /var/www/

Éditez le fichier de configuration du prelude-php-frontend (**/var/www/prelude-php-frontend/config.php**), et y écrire où décommenter les lignes

suivantes :

$server[1]['description'] = "SYSDOOR/MySQL phpfront v".VERSION ;
$server[1]['dbtype'] = USE_DB_MYSQL
$server[1]['dbusername'] = "prelude"
$server[1]['dbpassword'] = "dessstri"
$server[1]['dbhostname'] = LOCAL_CONNECTION ;
$server[1]['dbport'] = DEFAULT_PORT
$server[1]['dbname'] = "prelude"

Éditez le fichier **/var/www/prelude-php-frontend/index.php**, et y modifier la ligne

$serv= 0 en $serv = 1

Relancer le serveur web pour prendre en compte les changements :
/etc/init.d/apache-ssl restart
L'interface est maintenant accessible par **https ://localhost/prelude-php-frontend/.**

1.2.17 Installation et configuration du prelude-perl-frontend

1.2.18 Installation préalable de paquetages

Il faut bien sûr que perl et quelques autres modules soient installés. Pour cela, tapez la commande suivante :

apt-get install libdbi-perl libgd-graph-perl libdate-calc-perl
apache-ssl

1.2.19 Installation de prelude-perl-frontend

Télécharger les sources sur **www.leroutier.net/Projects/**
Nous utiliserons ici les sources suivantes :

prelude-perl-web-frontend.tar.gz

On les décompresse dans le répertoir /var/www/frontend-perl .

Dans le fichier de configuration d'Apache, il y a entre autres :

User www-data
Group www-data

Ils définissent le profil utilisateur d'Apache. On va changer les droits des fichiers du frontend-perl selon ces paramètres :

chown -R www-data.www-data /var/www/frontend-perl

chmod -R u+x /var/www/frontend-perl/

1.2.20 Configuration d'Apache

Modifez le fichier **/etc/apache-ssl/httpd.conf**, pour avoir les lignes suivantes : Listen192.168.0.2

DirectoryIndex index.pl index.html index.htm
ServerName localhost
DocumentRoot /var/www/
et rajouter les lignes suivantes :
<Directory "/var/www/frontend-perl/">
Options ExecCGI
AddHandler cgi-script .pl
</Directory>
Puis transformer

 # If the perl module is installed, this will be enabled.

<IfModule mod_perl.c>
Alias /perl/ /var/www/perl/
<Location /perl>
SetHandler perl-script
PerlHandler Apache::Registry
Options +ExecCGI
</Location>
</IfModule>

en ce qui suit

If the perl module is installed, this will be enabled.

<IfModule mod_perl.c>
Alias /perl/ /var/www/frontend-perl/
<Location /var/www/frontend-perl/>
SetHandler perl-script
PerlHandler Apache::Registry
Options +ExecCGI
</Location>
</IfModule>

- **Configuration de prelude-perl-frontend**

Éditer le fichier **/var/www/frontend-perl/Functions/config.p**l pour y faire apparaître les lignes suivantes :

$conf{’dbtype’}=’mysql’;
$conf{’dbname’}=’prelude’;
$conf{’dbhost’}=’localhost’;
$conf{’dbport’}=3306; # default mysql port is 3306
$conf{’dblogin’}=’prelude’;
$conf{’dbpasswd’}=’dessstri’;

L’interface est maintenant accessible par **https ://localhost/frontend-perl/** avec le login prelude et le mot-de-passe dessstri.

3. Conclusion

Beaucoup d'IDS sont fiables, ce qui entraîne leur intégration quasi systématique dans les solutions de sécurité. Les IDS présentent des avantages par rapport aux autres outils, mais aussi des inconvénients, des apports ainsi que des lacunes, qui en font des outils indispensables et non suffisants.
La particularité des systèmes de détection d'intrusions est d'apporter une sécurité non automatisée. Nous entendons par cela que la mise en place doit toujours s'accompagner d'une expertise sécurité dans les traitements des alertes remontées.
En ce qui concerne l'environnement d'étude, la particularité de travailler sur un projet libre, alors qu'il se trouve dans une période très active, est très enrichissante .Pour finir, nous dirons que tout ce travail a été très enrichissant techniquement et dans la méthodologie de travail, car il nous a ouvert un peu plus les portes de la sécurité informatique, et plus particulièrement sur les IDS. Cette vaste étude nous permet d'affirmer que nous possédons maintenant de bonnes compétences sur les Systèmes de Détection d'Intrusion ... et que Prelude-IDS et Snort sont tous deux de bons outils avec leurs propres particularités.

2. Webographie
1. Installation et configuration de Prelude-IDS

Site officiel de Prelude-IDS : http ://www.prelude-ids.org
Dernière version du document d'installation de Prelude-IDS : http ://lehmann.free.fr
2. Téléchargement des paquetages de Prelude-IDS :

http ://www.prelude-ids.org/rubrique.php3 ?id_rubrique=6
Autres documentations à propos de Prelude-IDS :
http ://www.prelude-ids.org/rubrique.php3 ?id_rubrique=1
Site officiel du frontend perl : http ://www.leroutier.net/Projects/
Site de Debian : http ://www.debian.org
Se procurer Debian sur CD : http ://ikarios.com/form/
3. Aide sur les Systèmes de Détection d'Intrusion
http ://www.securiteinfo.com : Site sur la sécurité informatique. On peut y trouver des explications
intéressantes sur les différentes attaques existantes.
http ://www.snort.org/docs/idspaper/ : Document présentant les faiblesses des IDS.
http ://www.secusys.com : Site sur la sécurité informatique. Contient quelques documents intéressant
lors d'un tout premier contact avec ce domaine.
Misc Num 3 : Magazine sur la sécurité informatique. Le numéro 3 contient un dossier spécial sur les IDS.
www.security-labs.org/index.php3 ?page=408
www.securiteinfo.com/conseils/choix_ids.shtml
www.securite.org/db/securite/ids/outils
http ://abcdelasecurite.free.fr/html/modules.php ?name=Downloads

4. Bibliographie pour intégrer les règles de Snort à Prelude-NIDS
Site officiel de Prelude-IDS : http ://www.prelude-ids.org
Site officiel de snort : http ://www.snort.org
Site où l'on peut télécharger le script convert_ruleset :
http ://mops.uci.agh.edu.pl/ kzaraska/prelude-contribs/
Téléchargement des paquetages de Prelude-IDS :
http ://www.prelude-ids.org/rubrique.php3 ?id_rubrique=6

5.Téléchargement des règles de Snort : http ://www.snort.org/dl/rules/

6. Projets liés de Prelude-IDS
Site officiel du projet Trithème : http ://tritheme.sourceforge.net/
Bibliographie sur Honeyd
http ://www.citi.umich.edu/u/provos/honeyd/
http ://www.citi.umich.edu/u/provos/papers/honeyd-dfn/
http ://www.securityfocus.com/infocus/1659

http ://www.citi.umich.edu/u/provos/systrace/
http ://www.onlamp.com/pub/a/bsd/2003/01/30/Big_Scary_Daemons.html
Bibliographie sur Nessus
Site officiel de Nessus : http ://www.nessus.org
Corrélation entre Prelude et Nessus : http ://www.rstack.org/oudot/prelude/correlation/
Bibliographie sur Logcheck
Fiche sur logcheck : http ://www.micro-fun.ch/techniques/fiches/controle_logs.shtml

5. GLOSSAIRE

Bibliothèque Fichier regroupant un ensemble de fonctions utilisable par plusieurs programmes

DoS : Denis Of Service : classe d'attaque visant a empêcher le système cible effectuer le service pour lequel il a été conçu.

Faux Positif Fausse alerte remontée par le système de détection d'intrusion.

FTP File Transfert Protocole : Protocole dédie au transfert de fichier.

HIDS Host Intrusion Detection System : Système de Detection d'intrusion oriente hôte, par opposition aux systèmes orientes réseau.

Kernel Voir Noyau

Noyau Coeur d'un système d'exploitation, il fait le lien entre le matériel et les applications. Permet entre autre de gérer le multitâche.

NIDS Network Intrusion Detection System : Système de detection d'intrusion oriente réseau

Open Scan Scan basique déterminant un service ouvert par l'établissement d'une connexion et sa fermeture immédiate

.

OSI Open System Interconnections : base commune établie par l'ISO pour décrire les services fournis par un réseau informatique.

Pile Espace mémoire disponible pour chaque processus lui permettant entre autre de sauvegarder son état.

PKI Public Key Infrastructure : Infrastructure a clef publique, mettant en jeu les concepts de chiffrement asymétrique, de certification et signature numérique dans le but d'identifier et d'authentifier des entités

RFC Request for Comments : specification de normes de l'IETF

Scan Technique générale consistant `a rechercher des services réseaux disponibles sur une machines.

SSL (secure socket layer) est un protocole se sécurité des échanges réseau

SSH protocole destiné a permettre l'ouverture d'une session sécurisé a distance.

Sniffer Outil permettant d'écouter un traffic (réseau) qui n'est pas destine a la machine hôte.

TTL Time To Live : Champ de l'entête IP d'un paquet réseau définissant le nombre de saut(passage d'un noeud réseau a un autre) maximum effectuable avant que le paquet ne soit détruit.

URL Universal Remote Location :

UNIVERSITE CHEIKH ANTA DIOP DE DAKAR
FACULTE DES SCIENCES ET TECHNIQUES
DEPARTEMENT DE MATHEMATIQUES
ET INFORMATIQUE

MASTER 2 PROFESSIONNEL TRANSMISSION
DE DONNEES ET SECURITE DE L'INFORMATION

MEMOIRE DE MASTER II

Sujet

Etude comparative de deux systèmes de détection d'intrusion Snort & Prelude-NIDS

Présenté par : **ALZOUMA MAYAKI** Hassane
Le jeudi 05 octobre 2006

Devant le jury composé de :

Président : **Dr Mbaye Sene**
Membres :

Dr Cheih Theicouba	**Gueye**	Examinateur
Mr Mohamed Ould	**Deye**	Encadreur
Dr Djiby	**Sow**	Chargé du suivi des stages
Mr Mohamet	**Sall**	Examinateur

Résumé : Un IDS (Intrusion Detection System) est un élément de l'architecture de sécurité d'un réseau, ayant pour tache principale de détecter une éventuelle intrusion sur un réseau ou sur une machine.
On peut aujourd'hui classer les systèmes de détection d'intrusion selon deux grandes familles : HIDS (Host IDS) au niveau système et application, NIDS: (Network IDS) au niveau réseau.
Les IDS, de par leur rôle de surveillance des activités dans des équipements informatiques (réseaux, machines, ..) se révèlent indispensable pour tout dispositif de sécurité performant. Les IDS nécessitent une bonne maîtrise et une administration correcte pour être efficaces. Cela nous a amené à faire une étude poussée sur les IDS. Vu la multitude de version IDS sur le marché, il est important entre autre de pouvoir motivé le choix d'un IDS par une étude comparative avec d'autres IDS de caractéristiques similaires.
C'est dans ce contexte qu'il nous a été donné l'occasion

- de faire une synthèse d'un certain nombre d'études faites sur la comparaison de SNORT et de PRELUDE
- puis de mettre en place SNORT à des fins de tests

Pour la plateforme de test pour SNORT, nous avons considéré sous Linux (Debian noyau 2.4) une architecture composée : d'une machine firewall, d'une machine qui héberge la BD Mysql dans la zone DMZ, d'une machine sonde qui héberge SNORT et trois machines dont deux, jouant le rôle de client situés dans la partie LAN et un, celui de cible pour les attaques, et qui est situé dans la partie DMZ.
A partir des machines clients, nous avons lancé des attaques avec tcpdump et nmap vers la machine cible. Nous utilisé tcpdump et nmap car ce sont les outils les plus utilisés par les hackers.
La sonde SNORT a pu relever les alertes et les a stockées localement.
Vu que les architectures de test de SNORT et de PRELUDE sont identiques, nous n'avons pas fait les tests avec PRELUDE, mais néanmoins, nous avons proposé une démarche de mise en place.

En perspective nous nous proposons de faire les mêmes tests sous Windows. On peut déjà noté que le travail est similaire à celui fait dans ce mémoire

Mots clés : Comparaison, sécurité, IDS, SNORT, PRELUDE, firewall, réseaux, systèmes d'exploitations