République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur
et de la recherche scientifique
université ibn khaldoun-Tiaret

Faculté des sciences, de la technologie et science de la matière
Département informatique

Mémoire de fin d'études

Pour l'obtention du diplôme
de master 2 en Réseau & Telecom

OPTION : Réseau & Telecom

Type : professionnel

Par :

Mr. Bader Ahmed
Mr. Kebas Otmane

Thème :

Système de detection des intrusions

Dirigé Par : Mr .BAKKAR Khaled

Année universitaire : 2011/2012

Sommaire

Chapitre III : les bases d'IDS

Chapitre IV : Implémentation

Introduction générale

Sample output to test PDF Combine only

L'informatique et en particulier l'Internet jouent un rôle grandissant dans notre société. Un grand nombre d'applications critiques d'un point de vue de leur sécurité sont déployées dans divers domaines comme le domaine militaire, la santé, le commerce électronique, etc. La sécurité des systèmes informatiques devient alors une problématique essentielle tant pour les individus que pour les entreprises ou les états.

Pour chaque système informatique, une politique de sécurité doit être définie pour garantir les propriétés de sécurité qui doivent être rendues par ce dernier. Cette politique s'exprime par des règles fixant trois objectifs distincts :

- La confidentialité
- L'intégrité
- La disponibilité

Nous entendons par intrusion, une violation d'un de ces trois objectifs. Plusieurs approches ont été définies pour s'assurer que la politique de sécurité définie pour un système informatique est bien respectée. Elle peut en effet être contournée par un utilisateur malveillant ou plus simplement une faute de conception peut être à l'origine d'une violation.

La première approche s'appuie sur des mécanismes préventifs : il s'agit alors de mettre en place des dispositifs capables d'empêcher toute action qui entraînerait une violation de la politique de sécurité.

La deuxième approche pour traiter les intrusions consiste à détecter les violations de la politique de sécurité et à les signaler aux administrateurs pour qu'ils puissent prendre les mesures nécessaires pour remédier aux problèmes éventuels qu'ont pu générer ces violations.

Une troisième approche, la tolérance aux intrusions, vise à garder le service assuré et que la politique de sécurité du système global reste inviolée même en présence d'intrusions dans certains composants du système. Des intrusions peuvent affecter certains composants du système mais les propriétés de confidentialité, d'intégrité et de disponibilité du système global doivent être vérifiées.

Les travaux que nous présentons dans cette mémoire en présenté trois chapitres fondamentale, dans le première chapitre on a parlé de quelque type d'attaque et dans le deuxième chapitre présentons les systèmes de détection d'intrusion et dans le troisième chapitre évaluation d'un ID

Le présent travail est repartie en quatre chapitre, au premier chapitre on traite le défirent de système d'information. Et on présent ou deuxième chapitre les systèmes de detection d'intrusion. Et en troisième chapitre ou se montre la valeur ajoute de notre recherche ou on a procéder les base d IDS (analyse de protocole et gère les alerte via de règle).Et en fin l'implémentation de notre application.

Chapitre I

L'attaque d'un réseau

Sample output to test PDF Combine only

Introduction

Les systèmes d'information sont aujourd'hui de plus en plus ouverts sur Internet. Cette ouverture, a priori bénéfique, pose néanmoins un problème majeur : il en découle un nombre croissant d'attaques

I. 1. Définition d'une attaque :

C'est une Action malveillante qui tente d'exploiter une faiblesse dans le système et de violer un ou plusieurs besoins de sécurité

I. 2. Anatomie d'une attaque :

Une attaque est appelée Habituellement « les 5 P » dans la littérature, ces cinq verbes anglophones : Probe, Penetrate, Persist, Propagate, Paralyze

- **Probe :** la collecte d'informations sur le système cible peut s'effectuer de plusieurs manières, comme par exemple un scan des ports grâce au programme Nmap pour déterminer la version des logiciels utilisés, et des outils comme firewalk, hping ou SNMP Walk permettent quant à eux de découvrir la nature d'un réseau

- **Penetrate :** utilisation des informations récoltées pour pénétrer un réseau. Des techniques comme le brute force ou les attaques par dictionnaires peuvent être utilisées pour outrepasser les protections par mot de passe.
- **Persist :** création d'un compte avec des droits de super utilisateur pour pouvoir se ré infiltre ultérieurement. Une autre technique consiste à installer une application de contrôle à distance capable de résister à un reboot
- **Propagate :** cette étape consiste à observer ce qui est accessible et disponible sur le réseau local.

- **Paralyze** : cette étape peut consister en plusieurs actions. Le pirate peut utiliser le serveur pour mener une attaque sur une autre machine, détruire des données ou encore endommager le système d'exploitation dans le but de planter le serveur

I. 3. Différent types d'attaques

I. 3.1. Les attaques distantes : Les attaques distantes, quant à elles, permettent à toute personne potentiellement raccordée à votre réseau de tenter sa chance, ce qui, dans le cas d'Internet peut devenir un réel problème, il existe deux types

A. Les attaques directes :

C'est la plus simple des attaques. Le hacker attaque directement sa victime à partir de son ordinateur

B. Les attaques indirectes :

Le hacker attaque indirectes sa victime via un 'ordinateur intermédiaire pour masquer l'identité et utiliser les ressources de l'ordinateur intermédiaire car il est plus puissant.

I. 3.2. Les attaques locales « applicatives »: Les attaques locales peuvent être dûes à un logiciel défectueux ou une mauvaise configuration de votre système. Dans le premier cas, deux possibilités existent :

a. Une faille logicielle existante mais non connue du grand public la seule solution est la veille passive sur l'exécution des processus de chaque utilisateur.

b. Une faille logicielle existante connue du grand public se présente généralement comme portillon, et qui, suite à une maintenance défectueuse permet à la personne mal intentionnée.

I. 4. Les attaques d'un réseau

Les attaques réseaux s'appuient sur des vulnérabilités liées directement aux protocoles ou à leur implémentation. Il en existe un grand nombre. Mais aujourd'hui il y a cinq attaques réseaux les plus connues

I. 4.1. Les techniques de scan

Les scans de ports ne sont pas des attaques à proprement parler. Le but des scans est de déterminer quels sont les ports ouverts, et donc en déduire les services qui sont exécutés sur la machine cible.

Il existe un nombre important de techniques de scan. Idéalement, la meilleure technique de scan est celle qui est la plus furtive afin de ne pas alerter les soupçons de la future victime. Voici une description des techniques de scan les plus répandues :

1) **Le scan simple** : aussi appelé le scan connect, il consiste à établir une connexion TCP complète sur une suite de ports. S'il arrive à se connecter, le port est ouvert ; sinon, il est fermé. Cette méthode de scan est très facilement détectable.

2) **Le scan furtif :** aussi appelé scan SYN, il s'agit d'une amélioration du scan simple. Ce scan essaie également de se connecter sur des ports donnés, mais il n'établit pas complètement la connexion : pas de commande ACK (acquittement) après avoir reçu l'accord de se connecter. Grâce à ceci, la méthode est bien plus furtive que le scan normal.

3) **Les scans XMAS, NULL et FIN** : se basent sur des détails de la RFC du protocole TCP pour déterminer si un port est fermé ou non en fonction de la réaction à certaines requêtes. Ces scans sont moins fiables que le scan SYN mais ils sont un peu plus furtifs. La différence entre ces trois types de scan se situe au niveau des flags TCP utilisés lors de la requête.

4) **Le scan à l'aveugle** : s'effectue via une machine intermédiaire et avec du spoofing

Le système attaqué pense que le scan est réalisé par la machine intermédiaire et non par le pirate.

5) **Le scan passif** : est la méthode la plus furtive. Consiste à analyser les champs d'en-tête des paquets et les comparer avec une base de signatures qui pourra déterminer les applications qui ont envoyé ces paquets.

I. 4.2. Fragments attacks :

But : le but de cette attaque est de passer outre les protections des équipements de filtrage IP.

- ✓ **Finalité** : en passant outre les protections, un pirate peut par exemple s'infiltrer dans un réseau pour effectuer des attaques ou récupérer des informations confidentielles.
- ✓ **Déroulement** : deux types d'attaque sur les fragments IP peuvent être distingués.

I. 4.3. Tiny Fragments :

D'après la RFC791 (IP), tous les nœuds Internet (routeurs) doivent pouvoir transmettre des paquets d'une taille de 68 octets sans les fragmenter d'avantage. En effet, la taille minimale de l'en-tête d'un paquet IP est de 20 octets sans options. Lorsqu'elles sont présentes, la taille maximale de l'en-tête est de 60 octets. Le champ IHL (Internet Header Length) contient la longueur de l'en-tête en mots de 32 bits. Ce champ occupant 4 bits, le nombre de valeurs possibles vaut de 2^4 - 1 = 15 (il ne peut pas prendre la valeur 0000). La taille maximale de l'en-tête est donc bien 15*4 = 60 octets.

Enfin, le champ Fragment Offset qui indique le décalage du premier octet du fragment par rapport au datagramme complet est mesuré en blocs de 8 octets. Un fragment de données occupe donc au moins 8 octets. Nous arrivons bien à un total de 68 octets.

L'attaque consiste à fragmenter sur deux paquets IP une demande de connexion TCP. Le premier paquet IP de 68 octets ne contient comme données que les 8 premiers octets de l'en-tête TCP (ports source et destination ainsi que le numéro de séquence). Les données du second paquet IP renferment alors la demande de connexion TCP (flag SYN à 1 et flag ACK à 0).

Or, les filtres IP appliquent la même règle de filtrage à tous les fragments d'un paquet. Le filtrage du premier fragment (Fragment Offset égal à 0) déterminant cette règle elle s'applique donc aux autres (Fragment Offset égal à 1) sans aucune autre forme de vérification. Ainsi, lors de la défragmentation au niveau IP de la machine cible, le paquet de demande de connexion est reconstitué et passé à la couche TCP. La connexion s'établit alors malgré le filtre IP.

I. 4.4. IP Spoofing :

Le but de cette attaque est l'usurpation de l'adresse IP d'une machine. Ceci permet au pirate de cacher la source de son attaque (utilisée dans les dénis de services dont nous discuterons plus tard) ou de profiter d'une relation de confiance entre deux machines. Nous expliquerons donc ici cette deuxième utilisation de l'IP Spoofing.

Le principe de base de cette attaque consiste à forger ses propres paquets IP (avec des programmes comme hping2 ou nemesis) dans lesquels le pirate modifiera, entre autres, l'adresse IP source. L'IP
Spoofing est souvent qualifié d'attaque aveugle (ou Blind Spoofing). Effectivement, les réponses éventuelles des paquets envoyés ne peuvent pas arriver sur la machine du pirate puisque la source est falsifiée. Ils se dirigent donc vers la machine spoofée. Il existe néanmoins deux méthodes pour récupérer des réponses.

Le Source Routing :

Le protocole IP possède une option appelée Source Routing autorisant la spécification du chemin que doivent suivre les paquets IP. Ce chemin est constitué d'une suite d'adresses IP des routeurs que les paquets vont devoir emprunter. Il suffit au pirate d'indiquer un chemin, pour le retour des paquets, jusqu'à un routeur qu'il contrôle. De nos jours, la plupart des implémentations des piles TCP/IP rejettent les paquets avec cette option.

Le Reroutage :

Les tables des routeurs utilisant le protocole de routage RIP peuvent être modifiées en leur envoyant des paquets RIP avec de nouvelles indications de routage. Ceci dans le but de rerouter les paquets vers un routeur que le pirate maîtrise.

I. 4.5. TCP Session Hijacking

Le TCP Session Hijacking permet de rediriger un flux TCP. Un pirate peut alors outrepasser une protection par un mot de passe (comme telnet ou ftp). La nécessité d'une écoute passive (sniffing) restreint le périmètre de cette attaque au réseau physique de la cible. Avant de détailler cette attaque, nous expliquerons quelques principes fondamentaux du protocole TCP.

Nous ne dévoilerons pas ici les arcanes du protocole TCP, mais préciserons uniquement les points essentiels à la compréhension de l'attaque. L'en-tête TCP est constitué de plusieurs champs.

- le port source et le port destination, pour identifier la connexion entre deux machines;
- le numéro de séquence qui identifie chacun des octets envoyés;
- le numéro d'acquittement qui correspond au numéro d'acquittement du dernier octet reçu;
- les flags, avec ceux qui vont nous intéresser sont :
  1. SYN qui synchronise les numéros de séquence lors de l'établissement d'une connexion;
  2. ACK, le flag d'acquittement d'un segment TCP;
  3. PSH qui indique au récepteur de remonter les données à l'application

I. 4.6. DNS Spoofing

But : fournir de fausses réponses aux requêtes DNS, c'est-à-dire indiquer une fausse adresse IP pour un nom de domaine.

- ✓ **Finalité :** rediriger, à leur insu, des Internautes vers des sites pirates. Grâce à cette fausse redirection, l'utilisateur peut envoyer ses identifiants en toute confiance par exemple

- ✓ **Déroulement** : il existe deux techniques pour effectuer cette attaque.

A. DNS Cache Poisoning :

Les serveurs DNS possèdent un cache permettant de garder pendant un certain temps la correspondance entre un nom de machine et son adresse
IP. Le DNS Cache Poisoning consiste à corrompre ce cache avec de fausses informations. Ces fausses informations sont envoyées lors d'une réponse d'un serveur
DNS contrôlé par le pirate à un autre serveur DNS, lors de la demande de l'adresse IP d'un domaine le cache du serveur ayant demandé les informations est alors corrompu.

B. *D*NS ID Spoofing :

Pour communiquer avec une machine, il faut son adresse IP. On peut toutefois avoir son nom, et grâce au protocole DNS, nous pouvons obtenir son adresse IP. Lors d'une requête pour obtenir l'adresse IP à partir d'un nom, un numéro d'identification est placé dans la trame afin que le client et le serveur puissent identifier la requête. L'attaque consiste ici à récupérer ce numéro d'identification (en sniffant le. réseau) lors de la communication entre un client et un serveur DNS, puis, envoyer des réponses falsifiées au client avant que le serveur DNS lui réponde **.**

I. 4.7. ARP Spoofing

Cette attaque, appelée aussi ARP Redirect, redirige le trafic réseau d'une ou plusieurs machine vers la machine du pirate. Elle s'effectue sur le réseau physique des victimes. Au préalable nous ferons un rappel sur l'utilité et le fonctionnement du protocole ARP. Le protocole ARP (Address Resolution Protocol) implémente le mécanisme de résolution d'une adresse IP en une adresse MAC Ethernet. Les équipements réseaux communiquent en échangeant des trames Ethernet (dans le cas d'un réseau Ethernet bien sûr) au niveau de la couche liaison de données. Pour pouvoir échanger ces informations il est nécessaire que les cartes réseau possèdent une adresse unique au niveau Ethernet, il s'agit de l'adresse MAC (Media Access Control). Quand un paquet IP doit être envoyé la machine expéditrice a besoin de l'adresse MAC du destinataire. Pour cela une requête ARP en broadcast est envoyée à chacune des machines du réseau physique local.

Cette requête pose la question : "Quelle est l'adresse MAC associée à cette adresse IP. La machine ayant cette adresse IP répond via un paquet ARP, cette réponse indiquant à la machine émettrice l'adresse MAC recherchée. Dès lors, la machine source possède l'adresse MAC correspondant à l'adresse IP destination des paquets qu'elle doit envoyer. Cette correspondance sera gardée pendant un certain temps au niveau d'un cache (pour éviter de faire une nouvelle requête à chaque paquet IP envoyé).

Cette attaque corrompt le cache de la machine victime. Le pirate envoie des paquets ARP réponse à la machine cible indiquant que la nouvelle adresse MAC correspondant à l'adresse IP d'une passerelle (par exemple) est la sienne. La machine du pirate recevra donc tout le trafic à destination de la passerelle, il lui suffira alors d'écouter passivement le trafic (et/ou le modifier). Il routera ensuite les paquets vers la véritable destination.

L'ARP Spoofing sert dans le cas où le réseau local utilise des switchs. Ceux-ci redirigent les trames Ethernet sur des ports différents selon l'adresse MAC. Il est dès lors impossible à un sniffer de capturer des trames au-delà de son brin physique. L'ARP Spoofing permet ainsi d'écouter le trafic entre des machines situées sur des brins différents au niveau du switch

I. 5. Les attaques applicatives :

Les attaques applicatives s'appuient principalement sur des vulnérabilités spécifiques aux applications utilisées. Cependant, certaines attaques peuvent être classées par type

I. 5.1. Les problèmes de configuration

Un des premiers problèmes de sécurité engendré par les applications est celui des erreurs de configurations. Nous distinguerons deux types d'erreurs : les installations par défaut et les mauvaises configurations à proprement parler.

Des logiciels, comme les serveurs Web, installés par défaut ont souvent des sites exemples qui peuvent être utilisés par des pirates pour accéder à des informations confidentielles. Par exemple, il peut y avoir des scripts permettant d'obtenir les sources des pages dynamiques ou des informations sur le système utilisé. En outre, lors d'une telle installation une interface d'administration à distance est disponible avec un login/mot de passe par défaut (trouvable dans le guide d'administration de l'application). Le pirate a donc la main sur le site et peut le modifier selon son bon vouloir

I. 5.2. Les "bugs"

Une mauvaise programmation des logiciels entraîne obligatoirement des "bugs". Ceux-ci seront la source des failles de sécurité les plus importantes. Ces vulnérabilités quand elles sont découvertes vont permettre d'exécuter des commandes non autorisées, obtenir la source de pages dynamiques, rendre indisponible un service, prendre la main sur la machine, etc. Les plus connus de ces "bugs" et les plus intéressants en ce qui concerne leur exploitation sont les buffers overflow.

I. 5.3. Les scripts

Une mauvaise programmation des scripts a souvent une répercutions sur la sécurité d'un système. En effet, il existe des moyens d'exploiter des failles de scripts développés en Perl qui permettront de lire des fichiers hors racine Web ou d'exécuter des commandes non autorisées.

I. 5.4. Man in the middle :

L'objectif principal de cette attaque est de détourner le trafic entre deux machines. Cela pour intercepter, modifier ou détruire les données transmises au cours de la communication. Cette attaque est plus un concept qu'une attaque à part entière. Il existe plusieurs attaques mettant en œuvre ce principe du Man in The Middle, comme le DNS Man in the Middle qui qu'une utilisation du DNS Spoofing pour détourner le trafic entre un client et un serveur Web. De même, une application récente a été élaborée pour détourner du trafic SSH.

I. 5.5. Les dénis de service :

Cette attaque porte bien son nom puisque qu'elle aboutira à l'indisponibilité du service (application spécifique) ou de la machine visée. Nous distinguerons deux types de déni de services, d'une part ceux dont l'origine est l'exploitation d'un bug d'une application et d'autre part ceux dus à une mauvaise implémentation d'un protocole ou à des faiblesses de celui-ci.

I. 5.5.1. Les dénis de service applicatifs

Tout comme les vulnérabilités d'une application entraînent la possibilité de prendre le contrôle d'une machine (exemple du buffer overflow), elles peuvent aussi amener à un déni de service.

L'application sera alors indisponible par saturation des ressources qui lui sont allouées ou un crash de celle-ci.

I. 5.5.2. Les dénis de service réseaux

Il existe différents types de déni de service utilisant les spécificités des protocoles de la pile TCP/IP

I. 5.5.2.1 SYN Flooding:

Exploite la connexion en 3 phases de TCP (Three Way Handshake : SYN / SYN-ACK / ACK). Le principe est de laisser un grand nombre de connexions TCP en attente. Le pirate envoie de nombreuses demandes de connexion (SYN), reçoit les SYN-ACK mais ne répond jamais avec ACK. Les connexions en cours occupent des ressources mémoire, ce qui va entraîner une saturation et l'effondrement du système

I. 5.5.2.2. UDP Flooding

Ce déni de service exploite le mode non connecté du protocole UDP. Il crée un "UDP Packet Storm" (génération d'une grande quantité de paquets UDP) soit à destination d'une machine soit entre deux machines. Une telle attaque entre deux machines entraîne une congestion du réseau ainsi qu'une saturation des ressources des deux hôtes victimes. La congestion est plus importante du fait que le trafic

UDP est prioritaire sur le trafic TCP. En effet, le protocole TCP possède un mécanisme de contrôle de congestion, dans le cas où l'acquittement d'un paquet arrive après un long délai, ce mécanisme adapte la fréquence d'émission des paquets TCP, le débit diminue. Le protocole UDP ne possède pas ce mécanisme, au bout d'un certain temps le trafic UDP occupe donc toute la bande passant n'en laissant qu'une infime partie au trafic TCP

.

I. 5.5.2.3. Packet Fragment

Les dénis de service de type Packet Fragment utilisent des faiblesses dans l'implémentation de certaines pile TCP/IP au niveau de la défragmentation IP (réassemblage des fragments IP).

Une attaque connue utilisant ce principe est Teardrop. L'offset de fragmentation du second fragment est inférieur à la taille du premier ainsi que l'offset plus la taille du second. Cela revient à dire que le deuxième fragment est contenu dans le premier (overlapping).

Lors de la défragmentation certains systèmes ne gèrent pas cette exception et cela entraîne un déni de service.

Il existe des variantes de cette attaque : bonk, boink et newtear par exemple. Le déni de service Ping of Death exploite une mauvaise gestion de la défragmentation au niveau ICMP, en envoyant une quantité de données supérieure à la taille maximum d'un paquet IP. Ces différents dénis de services aboutissent à un crash de la machine cible.

I. 5.5.2.4. Smurfing :

Cette attaque utilise le protocole ICMP. Quand un ping (message ICMP ECHO) est envoyé à une adresse de broadcast (par exemple 10.255.255.255), celui-ci est démultiplié et envoyé à chacune des machines du réseau. Le principe de l'attaque est de spoofer les paquets ICMP ECHO REQUEST envoyés en mettant comme adresse IP source celle de la cible. Le pirate envoie un flux continu de ping vers l'adresse de broadcast d'un réseau et toutes les machines répondent alors par un message ICMP ECHO REPLY en direction de la cible. Le flux est alors multiplié par le nombre d'hôte composant le réseau. Dans ce cas tout le réseau cible subira le déni de service, puisque l'énorme quantité de trafic généré par cette attaque entraîne une congestion du réseau.

I. 5.5.3. Déni de service distribué :

Le but est ici de reproduire une attaque normale à grande échelle. Pour ce faire, le pirate va tenter de se rendre maître d'un nombre important de machines. Grâce à des failles (buffer overflows, failles RPC …..… etc.) .Il va pouvoir prendre le contrôle de machines à distance et ainsi pouvoir les commander à sa guise.

Une fois ceci effectué, il ne reste plus qu'à donner l'ordre d'attaquer à toutes les machines en même temps, de manière à ce que l'attaque soit reproduite à des milliers d'exemplaires. Ainsi, une simple attaque comme un SYN Flooding pourra rendre une machine ou un réseau totalement inaccessible

Sommaire

Chapitre II

les systèmes de detection d'intrusion

Introduction

Aucun système d'information n'est sûr à cent pour cent contre les menaces les intrusions et Parmi les préceptes connus sur la sécurité informatique se trouve celui énonçant que, pour une entreprise connectée à l'Internet, le problème aujourd'hui n'est plus de savoir *si* elle va se faire attaquer, mais quand cela va arriver, la solution possible est alors d'essayer de repousser les risques dans le temps par la mise en œuvre de divers moyens destinés à augmenter le niveau de sécurité. Pour contrer les menaces d'intrusion, les entreprises se tournent de plus en plus vers les solutions de détection d'intrusion, dont les possibilités faramineuses sont vantées par les sociétés éditrices de ces logiciels. Mais le décalage entre le discours commercial et les possibilités techniques réelles de ces produits peut être important, et les conséquences fâcheuses lorsqu'il s'agit de sécurité de l'information.

II. 1. Détection d'intrusion

La détection d'intrusion est l'acte de détecter les actions qui essaient de compromettre la confidentialité, l'intégrité ou la disponibilité d'une ressource.

La détection d'intrusion peut être effectuée manuellement ou automatiquement. Dans le processus de détection d'intrusion manuelle, un analyste humain procède à l'examen de fichiers de logs à la recherche de tout signe suspect pouvant indiquer une intrusion.

Un système qui effectue une détection d'intrusion automatisée est appelé système de détection d'intrusion (IDS).

II. 2. Les systèmes de détection d'intrusions (IDS)

II. 2.1. Définition :

Un système de détection d'intrusion est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (réseau, hôte).

II. 2.2. Les techniques de détection d'intrusion.

II. 2.2.1. La détection d'abus (misuse detection)

Aussi appelée détection de mauvaise utilisation, l'IDS analyse l'information recueillie et la compare avec une base de données de signatures d'attaques connues toute activité correspondante est considérée comme une attaque.

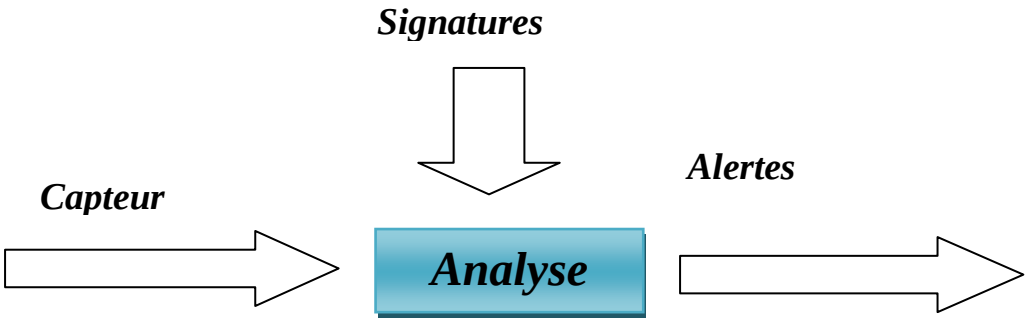


Figure I : la méthode de detection par signatures

Avantage

1. Simplicité de mise en œuvre.
2. Rapidité de diagnostic.
3. Précision (en fonction des règles).
4. Identification du procédé d'attaque.

Inconvénients

1. Ne détecte que les attaques connues.
2. Maintenance de la base.

II. 2.2.2. la détection d'anomalie (anomaly detection)

La détection d'anomalie de comportement est une technique assez ancienne elle est utilisée également pour détecter des comportements suspects en téléphonie, Cette technique basée sur le comportement « normal » du système

- Une déviation par rapport à ce comportement est considérée suspecte.
- Le comportement doit être modélisé on définit alors un profil.
- Une attaque peut être détectée sans être préalablement connue.

Avantages

1. Permet la détection d'attaque inconnue.
2. Facilite la création de règles adaptées à ces attaques.
3. Difficile à tromper.

Inconvénients

1. Les faux-positifs sont nombreux
2. Générer un profil est complexe
3. Diagnostics long et précis en cas d'alerte

II. 3. Fonctionnement d'un IDS

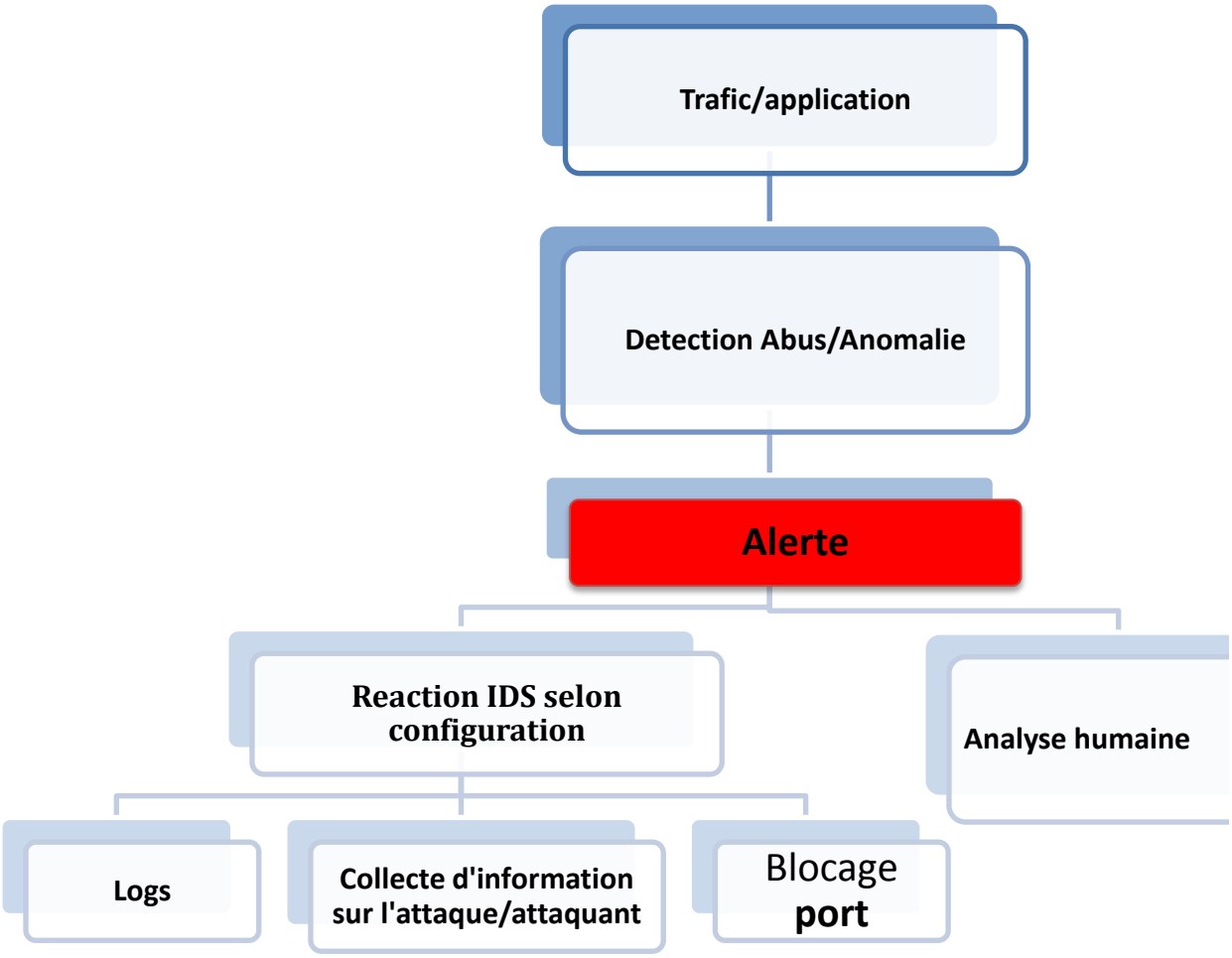


Figure 2. Illustre le fonctionnement d'un IDS.

II. 4. Utilité de l'IDS

Pourquoi nous avons besoin d'installer un IDS dans votre ordinateur ou de votre réseau? Pour surveiller la circulation des paquets sur le réseau. Vous pouvez considérez le IDS comme une caméra installé devant votre port. Ça pour savoir qui essaye à attaquer à votre réseau.

II. 5. Les types de système de détection d'intrusion

Les familles d'IDS et leurs variantes (localisation). Selon l'endroit qu'ils surveillent et ce qu'ils contrôlent (les sources d'information), deux familles principales d'IDS sont usuellement distinguées

- HIDS (*Host-based* Intrusion Detection System)
- NIDS (Network-based Intrusion Detection System)

II. 5.1. Le HIDS (*Host-based* Intrusion Detection System):

HIDS fait marcher sur les informations collectées à partir d'un système de l'ordinateur individuel. Cet avantage nous permet d'analyser des activités avec une grande fiabilité et précision, déterminant exactement quel processus et utilisateur sont concernés aux attaques particulières sur le système d'exploitation. De plus, HIDS peut surveiller les tentatives de la sortie, comme ils peuvent directement accéder et surveiller des données et des processus qui sont le but des attaques.

HIDS emploie normalement des sources de l'information de deux types, la traîné de l'audit du système d'exploitation et les journaux du système. La traîné de l'audit du système d'exploitation est souvent générée au niveau de noyau du SE, et elle est plus détaillé et plus protégé que les journaux du système. Pourtant les journaux est moins obtus et plus petit que la traîné de l'audit du SE, c'est ainsi qu'il est facile à comprendre.

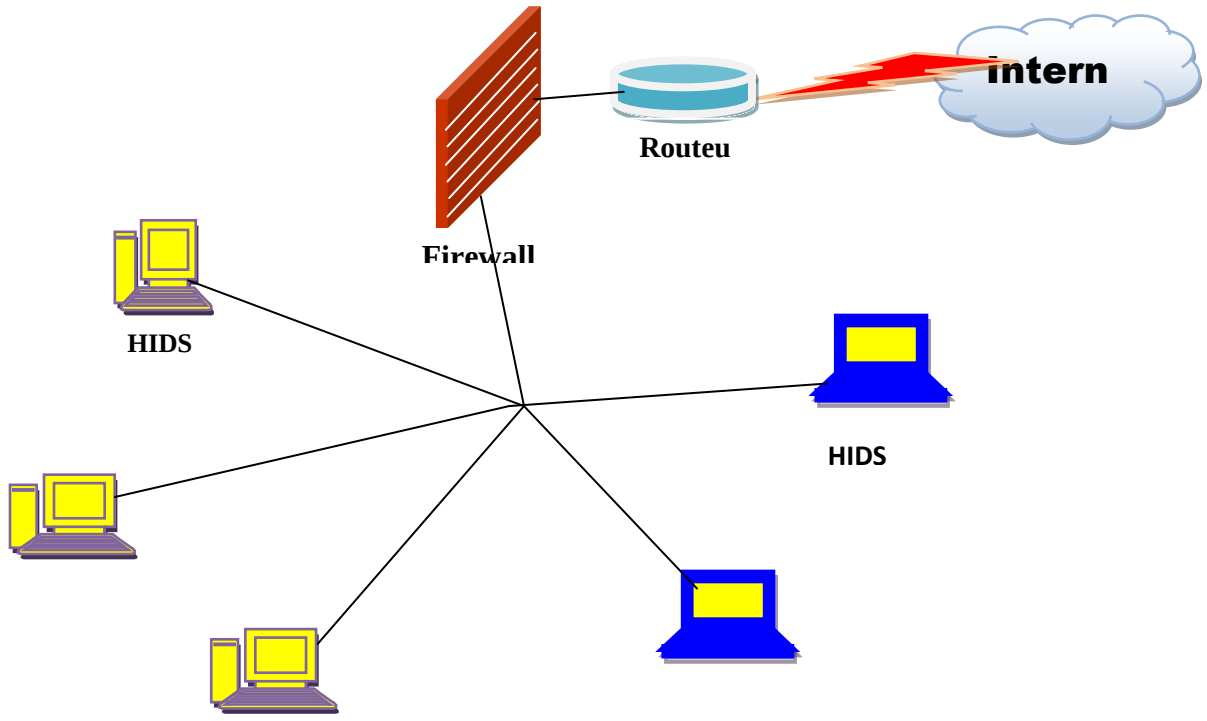


Figure 3 : Host-based Intrusion Detection System

Avantage :

1. Pouvoir surveiller des événements locaux jusqu'au host
2. Marcher dans un environnement dans lequel le trafic de réseau est encrypté
3. Ils peuvent détecter le Cheval de Troie ou les autres attaques concernant à la brèche intégrité de logiciel.

Inconvénients

1. HIDS est difficile à gérer, et doivent configurées et gérées pour chaque host surveillé
2. HIDS peut être neutralisé par certaine attaque de Dos
3. Lorsque HIDS emploie la traîné de l'audit du SE comme des sources des informations, la somme de l'information est immense, alors il demande le stockage supplémentaire local dans le système

II. 5.2. Le NIDS (Network-based Intrusion Detection System):

Un IDS réseau est un système de detection des intrusions travaille sur les trames réseau aux niveaux (couches réseau, transport, application), il est capable de détecter des paquets malveillants conçus pour outrepasser un pare-feu aux règles de filtrage trop laxistes, et de chercher des signes d'attaque à différents endroits sur le réseau.

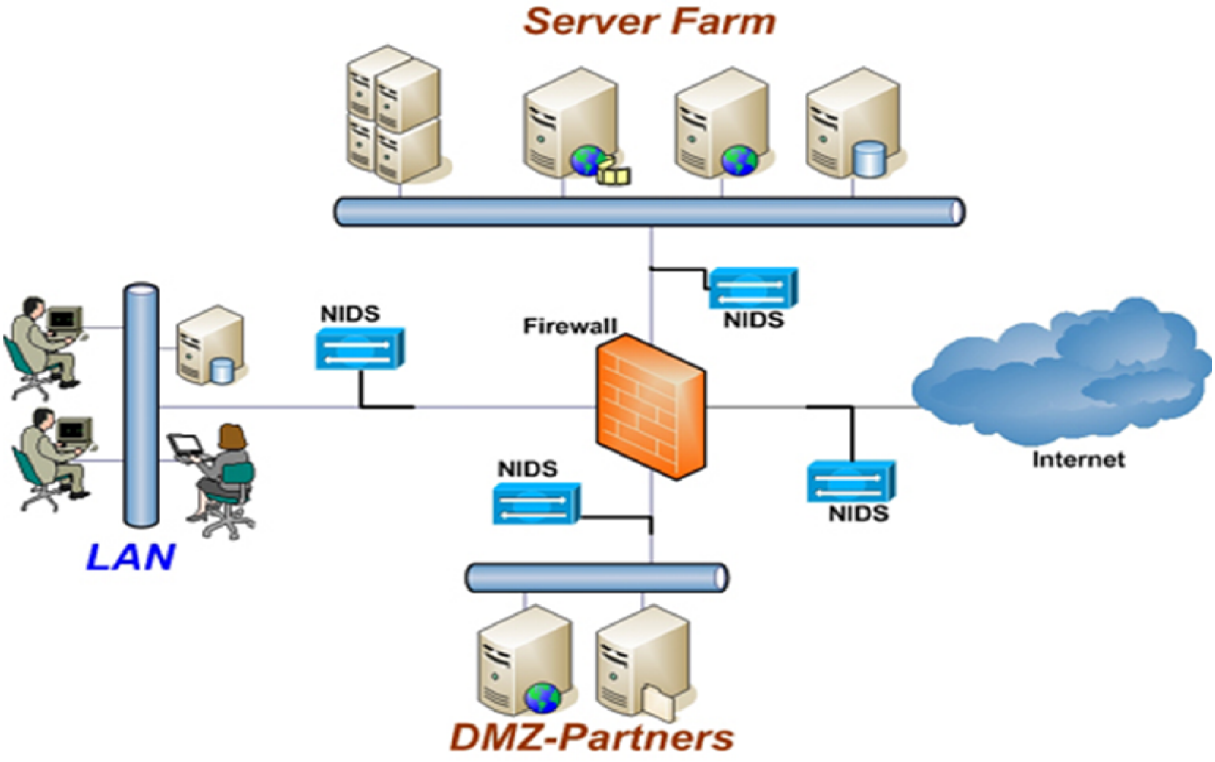


Figure 4 : NIDS Network-based Intrusion Detection System

.

Avantage :

1. Le NIDS peut surveiller un grand réseau.
2. L'déploiement de NIDS a peu d'impact sur un réseau existant.
3. NIDS peut être très sûr contre l'attaque et être même se cache à beaucoup d'attaquants

Inconvénients

1. Il est difficile à traiter tous les paquets circulant sur un grand réseau.
2. La plupart de NIDS ne peuvent pas indiquer si un attaque réussi ou non.
3. NIDS ne peut pas analyse des informations chiffrées

II. 5.3. Les systèmes de détection d'intrusions « hybrides » :

Généralement utilisés dans un environnement décentralisé, ils permettent de réunir les informations de diverses sondes placées sur le réseau. Leur appellation « hybride » provient du fait qu'ils sont capables de réunir aussi bien des informations provenant d'un système HIDS qu'un NIDS.

L'exemple le plus connu dans le monde Open-Source est prélude. Ce Framework permet de stocker dans une base de données et des alertes provenant de différents systèmes relativement variés. Utilisant Snort comme NIDS, et d'autres logiciels tels que Samhain en comme HIDS, il permet de combiner des outils puissants tous ensemble pour permettre une visualisation centralisée des attaques.

Même si je distingue HIDS et NIDS, la différence devient de plus en plus réduite puisque les HIDS possèdent maintenant les fonctionnalités de base des NIDS. Des IDS bien connus comme ISS Real Secure se nomment aujourd'hui "IDS hôte et réseau". Dans un futur proche la différence entre les deux systèmes deviendra de plus en plus faible.

II. 5.4. Système de Détection d'Intrusion de Nœud Réseau (NNIDS) :

Ce nouveau type (NNIDS) fonctionne comme les NIDS classiques, c'est-à-dire vous analysez les paquets du trafic réseau. Mais ceci ne concerne que les paquets destinés à un nœud du réseau (d'où le nom). Une autre différence entre NNIDS et NIDS vient de ce que le NIDS fonctionne en mode promiscuous, ce qui n'est pas le cas du NNIDS. Puisque tous les paquets ne sont pas analysés, les performances de l'ensemble sont améliorées.

II. 5.5. Détection d'Intrusion basée sur une Application

Les IDS basés sur les applications sont un sous-groupe des IDS hôtes, mais nous les mentir séparément. Ils contrôlent l'interaction entre un utilisateur et un programme ajoutant des fichiers de log afin de fournir des informations sur les activités. Puisque vous opérez entre utilisateur et programme, il est facile de filtrer tout comportement notable. Un ABIDS peut être visualisé la figure 5.

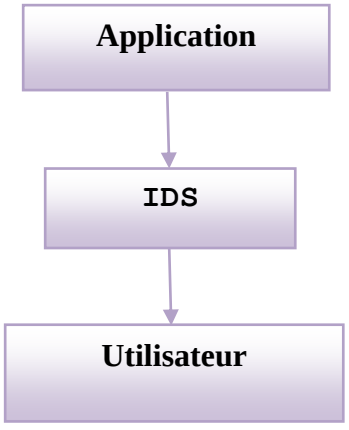


Figure 5 : IDS basée sur une Application

Avantage :

1. elle travaille en clair,
2. elle peut détecter et empêcher des commandes particulières dont l'utilisateur pourrait se servir avec le programme

Inconvénients :

1. faible sécurité et peu de possibilités de détecter, par exemple, un Cheval de Troie
2. les fichiers de log générés par ce type d'IDS sont cibles faciles pour les attaquants et ne sont pas aussi sûrs, par exemple, que les traces d'audit du système

II. 6. Les systèmes de prévention d'intrusions

II. 6.1 Définition :

Ensemble de composants logiciels et matériels dont la fonction principale est d'empêcher toute activité suspecte détectée au sein d'un système.

Contrairement aux IDS simples, les IPS sont des outils aux fonctions « actives », qui en plus de détecter une intrusion, tentent de la bloqué.

II. 6.2 Principes de fonctionnement

- La surveillance du comportement d'application se rapproche des IDS basés sur une application, c'est-à-dire que le comportement de l'application est analysé et noté (quelles données sont normalement demandées, avec quels programmes elle interagit, quelles ressources sont requises, etc.).
- La création de règles pour l'application : dérivé de la surveillance du comportement d'application, cet ensemble de règles donne des informations sur ce que peut faire ou non une application.
- La fonctionnalité d'alerte suite aux violations permet d'envoyer une alerte en cas de déviation (c'est-à-dire lorsqu'une attaque est détectée). L'alerte peut aller d'une simple entrée dans un journal à un blocage de ressources.

- L'interception d'appels au système : avant qu'un appel au système (rootkit) soit accepté, il doit être complètement vérifié Cette fonctionnalité permet la surveillance des essais de modification d'importants fichiers du système ou de la configuration.
- D'autres fonctionnalités sont possibles, comme la compréhension des réseaux IP (architecture, protocoles, etc.), la maîtrise des sondes réseau/analyse des logs, la défense des fonctions vitales du réseau, la vitesse d'analyse et un mode "stateful inspection". La prévention d'intrusion est une technique relativement nouvelle par comparaison aux autres techniques. Cette approche fait interagir des technologies hétérogènes

En peut réisme le fonctionnement d'un IPS dans le figure suivant :

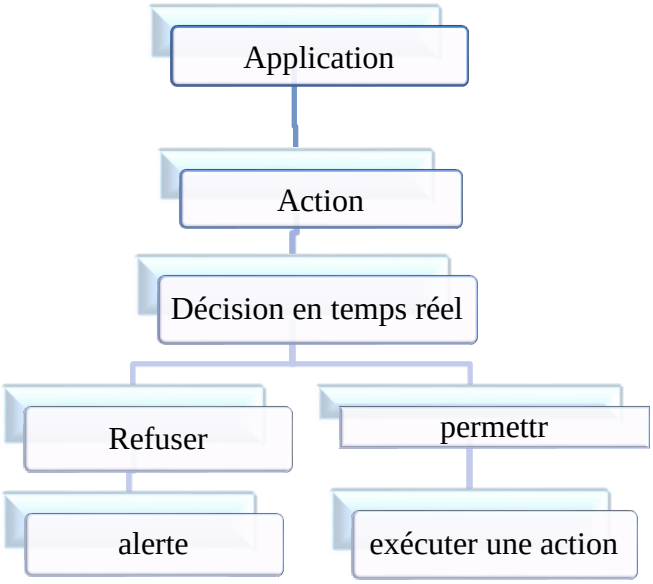


Figure 06 : le fonctionnement d'un IPS

II.6.3.Les objectifs d'un IPS :

- ❖ Interrompre une connexion.
- ❖ Ralentir la connexion.

Avantages

1. Attaque bloquée immédiatement.

Inconvénients

1. Les faux-positifs.
2. Peut paralyser le réseau.

II. 7. Les firewalls :

Les firewalls ne sont pas des IDS à proprement parler mais ils permettent également de stopper des attaques. Nous ne pouvions donc pas les ignorer.

Les firewalls sont basés sur des règles statiques afin de contrôler l'accès des flux. Ils travaillent en général au niveau des couches basses du modèle OSI (jusqu'au niveau 4), ce qui est insuffisant pour stopper une intrusion. Par exemple, lors de l'exploitation d'une faille d'un serveur Web, le flux HTTP sera autorisé par le firewall puisqu'il n'est pas capable de vérifier ce que contiennent les paquets.

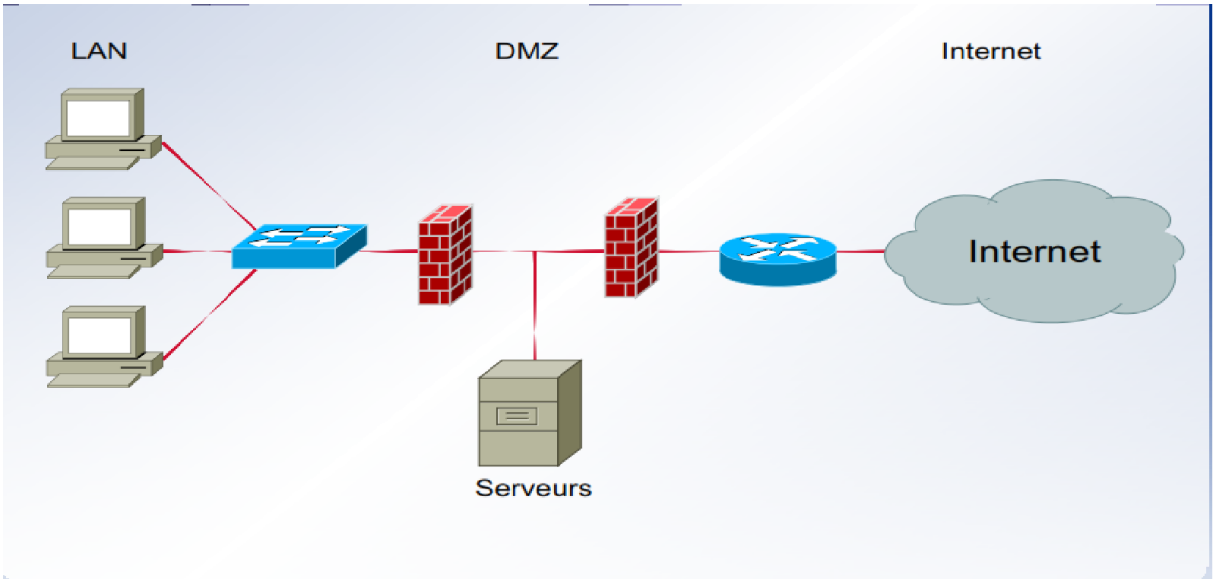


Figure 6 : firewalls

II. 7.1. Les différentes catégories de firewall

II. 7.1.1. Les systèmes à maintien d'état (stateful)

Vérifient que les paquets appartiennent à une session régulière. Ce type de firewall possède une table d'états où est stocké de chaque connexion établie, ce qui permet au firewall de prendre des décisions adaptées à la situation. Ces firewalls peuvent cependant être outrepassés en faisant croire que les paquets appartiennent à une session déjà établie.

Avantage :

1. Prends en charge l'état de la connexion TCP.
2. Vérifie la validité des paquets.
3. Acceptation de connexions.
4. Protège des attaques de types IP Spoofing et SYN Flood.

Inconvénient

1. Nécessite des ressources supplémentaires.

II. 7.1.2 Les systèmes à filtrage de paquets sans état :

Analyse les paquets les uns après les autres, de manière totalement indépendante.

Avantage :

1. Contrôle paquet par paquet.
2. Filtrage par adresses IP et par ports.

Inconvénient

1. Obligation d'ouvrir les ports 1024 pour les connexions vers l'extérieur.
2. Ne protège pas des attaques IP Spoofing et SYN flood.

II .8.Les domaines utilisation des IDS

II.8.1.Les banques et établissements financiers:

La propriété d'un peuple est dit que si elle est maintenue en banques. Seulement enregistrement numérique est donné à la personne tels que le numéro de compte, cartes, etc. pour accéder à son propriété. Si le format numérique est une fuite ou piraté, puis l'existence de banques ne peuvent pas être imaginé. Nous entendons dans les nouvelles de grande perte de valeur dans le piratage de codes de recharge nombre de société de téléphonie mobile. Donc le système de détection des intrusions sera certainement minimiser l'accès non autorisé et de prendre une réponse immédiate pour faire cesser cette travaux illégaux.

II.8.2.Entreprises multinationales:

Les entreprises multinationales seraient grâce à la gestion du programme. L'intégration des sociétés multinationales est due à toutes leurs transactions et des rapports sous forme numérique. Cette information est transmise à travers les différents réseaux. Cette transmise effectuées à travers des couches de sécurité sont ou différents Mais ils sont toujours vulnérables aux attaques par des pirates

Donc, le système de détection d'intrusion est très essentiel aux réseaux de réseaux informatiques tels sociétés multinationales.

Conclusion

Nous l'avons vu, les IDS sont des outils permettant de détecter les intrusions du réseau sur lequel il est placé. C'est un outil complémentaire aux outils de sécurisation d'un réseau base sur l'analyse des paquets, Ce dernier géré par des protocoles.

Pour décidé que le paquet intruse ou non il fat appliqué des règles qui bien détail dans le chapitre suivant.

Chapitre III

les bases des IDS

Introduction

Etant donné que nous allons traiter dans cette partie les différentes règles d'un IDS il parait nécessaire de rafraîchir quelques notions et plus particulièrement la notion de protocoles.

III. 1. Notion de protocole

La notion de protocole se retrouve à tous les niveaux du modèle en couche. Le sendmail par exemple s'appuie sur le protocole SMTP, les serveurs WWW sur le protocole HTTP.

Pour une couche donnée, on appelle protocole, l'ensemble des règles et des formats (sémantiques et syntaxiques) prédéfinis déterminant les caractéristiques de communication des processus de la couche. La mise en œuvre d'un protocole est effectuée à partir d'un PDU (Protocol Data Unit).

Lorsque l'on veut établir une communication, il est intuitivement indispensable de posséder trois informations Le nom de la machine distante, Son adresse, La route à suivre pour y parvenir.

III. 2. Les protocoles d'interconnections

- Le protocole IP
- Le protocole TCP
- Le protocole UDP
- Le protocole ICMP
- Le protocole HTTP
- Le protocole ARP
- Le protocole FTP
- Le protocole SSH
- Le protocole Telnet
- Le protocole pop3
- Le protocole SMTP

Dans cette partie nous décrivons en détaille les protocoles les plus utilisés dans les IDS

III. 2.1. Le protocole IP :

L'IP est le protocole spécifique à Internet, qui se charge de transmettre les données sous forme de paquets. L'envoi de ces paquets est réalisé en fonction des adresses de réseaux ou de sous-réseaux qu'ils contiennent.

Le protocole IP définit :

- L'unité de donnée transférée dans les interconnexions.
- la fonction de routage.
- les règles qui mettent en œuvre la remise de paquets en mode non connecté

Il est utile de connaître les champs du protocole IP, qui sont utilisés à des fins de reconnaissance ou d'attaque par déni de services.

<table>
<tr><td>Vers</td><td>Long entête</td><td>Type de service</td><td colspan="2">Longueur total</td></tr>
<tr><td colspan="3">Identificateur du fragment</td><td>Flags</td><td>Position du fragment</td></tr>
<tr><td colspan="2">Durée de vie</td><td>Protocole</td><td colspan="2">Contrôle d'erreur entête</td></tr>
<tr><td colspan="5">Adresse IP émetteur</td></tr>
<tr><td colspan="5">Adresse IP cible</td></tr>
<tr><td colspan="4">Option IP : longueur variable</td><td>A zéro alignement</td></tr>
<tr><td colspan="5">Donnée</td></tr>
</table>

Figure 1 : le paquet IP

Vers : numéro de version de protocole IP (IPv4 ou IPv6).

Long entête : longueur de l'en-tête en mots de 32 bits.

Longueur totale : Longueur totale (en-tête + données).

Type de service : indique comment doit être géré le datagramme.

Identificateur du fragment : entier qui identifie le datagramme initial

FLAGS : définit les déférents drapeaux utilisés dans l'entête IP

Position du fragment : Détermine la position d'un fragment dans un message il contient trois types

1. Position du fragment (OF)
2. Ne pas fragmenter (DF)
3. Dernier fragment (MF)

Durée de vie : Mesure du temps de séjour dans le réseau depuis l'émission en secondes

Protocole : Protocole qui utilise IP valeurs normalisées pour le démultiplexage des paquets entrants (TCP=6, UDP=17)

Contrôle d'erreur entête : Contrôle d'intégrité sur l'entête du paquet.

Adresse IP émetteur :

Adresse IP cible :

Option IP : Utilisée pour spécifier des compléments de protocole de 4 à 40 octets

Donnée : donnée utilisateur d'une taille maximum de 64 K octets.

III. 2.2. Le protocole TCP

TCP Transmission control protocole est un protocole chargé du contrôle lors du transfert de données. Son rôle consiste à vérifier que les paquets IP envoyés sont bien reçus en l'état, sans aucune perte ou changement sur le plan de leur intégrité.

Caractéristique :

Transmission de données en mode connecté par paquets de tailles variables.

L'échange de donnés est bidirectionnelle. .

Contrôle de la duplication et de flux et de congestion et de récupération des erreurs

Utilité de protocole TCP

- protection contre les erreurs,
- contrôle de congestion,
- contrôle de flux
- maintien de la séquentialité
- contrôle de la duplication

Les champs de TCP, comme ceux de IP, sont souvent utilisés par des pirates à des fins intrusives. Voici une description de ces champs :

<table>
<tr><td colspan="3">Port source</td><td>Port destination</td></tr>
<tr><td colspan="4">Numéro de séquence</td></tr>
<tr><td colspan="4">Numéro d'acquittement</td></tr>
<tr><td>Long entête</td><td>Réservé</td><td>Codes</td><td>Fenêtre</td></tr>
<tr><td colspan="3">Checksum</td><td>Pointeur urgence</td></tr>
<tr><td colspan="3">Options éventuelles</td><td>A zéro alignement</td></tr>
<tr><td colspan="4">Données</td></tr>
</table>

Figure 2 : le paquet TCP

Port source : numéro de port de la machine émetteur

Port destination : numéro de port de la machine récepteur

Numéro de séquence : est un numéro utilise pour numéroté l'octet a partir du premier octet

Numéro d'acquittement : le prochain numéro de séquence attendu par l'émetteur de cet acquittement

Réservé : usage futur ⟶ toujours à.

Code : indique la nature du segment

1. **URG** : le pointeur de données urgentes est valide
2. **SYN :** utilisé à l'initialisation de la connexion pour indiquer où la numérotation séquentielle commence
3. **FIN :** utilisé lors de la libération de la connexion
4. **PSH :** fonction Push
5. **RST :** utilisé pour réinitialiser la connexion

Fenêtre : la quantité de données que l'émetteur de ce segment est capable de recevoir
Checksum : calcul du champ de contrôle : utilise un pseudo-en-tête et s'applique à la totalité du segment obtenu
Pointeur urgence : indique la fin de la partie urgente du segment qui commence au début du champ de données du segment, Le champ "urgent pointer" est validé par le bit "urgent"
Options éventuelles : il existe plusieurs options

End of option List : permet à la partie variable de l'entête de se terminer en frontière de mot

MSS option : Permet de négocier la taille maximum des segments envoyés

A zéro alignement : Afin de réaliser l'alignement sur les mots de 32 bits, un bourrage ("padding") peut être nécessaire

III. 2.3. Le protocole UDP

UDP :User Datagram Protocole est protocole de transport sans connexion de service applicatif émission de messages applicatifs sans établissement de connexion. il Transmission les données par paquet en mode non connecté

Est protocole simple surcout minimal pour les paquets UDP et pour le traitement du protocole et pas de contexte,

- le service fourni est le service disponible.

- multiplexage (nº port).
- Adapté au multicast

Port UDP de l'émetteur	Port UDP cible
Longueur de message UDP	Checksum UDP
Données	

Figure 03 : Le paquet UDP

Format général :

Une entête de taille fixe.

Un champ de données de taille variable.

Longueur de message : longueur totale du paquet UDP<64 K octets (entête +données)

Port UDP de l'émetteur : il spécifie le nº de port utilisé lors de la question

Port UDP cible : il spécifie le nº de port utilisé lors de la réponse

Checksum UDP : Détection de la corruption du contenu du paquet UDP

Les numéros de port

Ces destinations abstraites permettant d'adresser un service applicatif s'appellent des ports de protocole

L'émission d'un message se fait sur la base d'un port source et un port destinataire

Les processus disposent d'une interface système leur permettant de spécifier un port ou d'y accéder.

III. 2.4. Le protocole ARP

L'ARP : Address Resolution Protocol est un protocole pour établir le lien entre adresse IP et l'adresse physique MAC

Rôle d'ARP : fournir à une machine donnée l'adresse physique d'une autre machine située sur le même réseau à partir de l'adresse IP de la machine destinatrice.

Diffusion d'adresse sur le réseau physique.

La machine émettrice émet un message contenant son adresse physique.

Type de matériel		Type de protocole
LGR-MAT	LGR-PORT	Opération
Adresse matériel émetteur		
Adresse matériel émetteur		Adresse IP émetteur
Adresse IP émetteur		Adresse matériel cible
Adresse matériel cible		
Adresse IP cible		

Figure 04 : ***Le paquet ARP***

Type de matériel : pour spécifier le type d'adresse physique

Type de protocole : pour spécifier le type d'adresse logique

LGR-MAT : pour spécifier la longueur de l'adresse physique

LGR-PORT : pour spécifier la longueur de l'adresse logique

Opération : pour précise le type de l'opération question ou réponse

La requête ARP est véhiculée dans un message protocolaire lui-même encapsulé dans la trame de liaison de données.

Lorsque la trame arrive à destination, la couche liaison de données détermine l'entité responsable du message encapsulé

La structure du message ARP gère une association adresse de protocole / adresse physique

III. 2.5. Le protocole ICMP

ICMP : Internet Control Message Protocol est un protocole qui permet d'envoyer des messages de contrôle ou d'erreur vers d'autres machines ou passerelles.et rapporte les messages d'erreur à l'émetteur initial.

- La machine destinatrice déconnectée
- La durée de vie du datagramme expirée
- lorsqu'un datagramme ne peut être acheminé vers sa destination
- lorsqu'un routeur arrive à saturation

Il est possible de générer une trame ICMP par la commande PING, Cette commande correspond à la recherche de la possibilité de mise en communication avec une machine d'adresse IP spécifie.

Fonctionnement

Le protocole ICMP est implémenté dans la couche IP en partie supérieure. Ainsi, il contrôle le fonctionnement de la couche IP, en s'appuyant sur elle pour la transmission de ses propres messages.

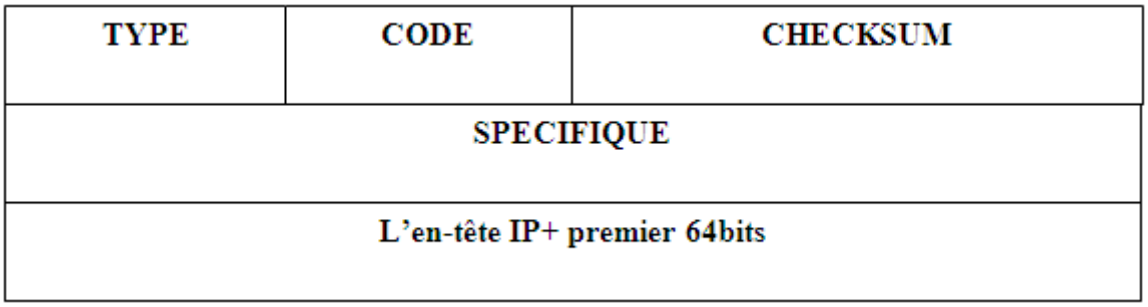


Figure 05 le message ICMP

TYPE : type de service ICMP par exemple

type	Signification
0	Réponse à une requête d'écho
3	Destination inaccessible
4	Limitation de débit de la source
5	Reroutage
10	Sélection de routeur
17	Requête de masque d'adresse
18	Réponse à une demande de masque d'adresse

CODE : indique le codage de l'erreur rapportée et est spécifique à chaque type d'erreur

SPECIFIQUE : est un champ de données spécifique au type d'erreur

III. 2.6. Le protocole FTP

FTP File Transfer Protocol :est un protocole applicatif pour le transfert de fichiers sur Internet. FTP comprend les fonctions nécessaires pour accéder au réseau, pour obtenir la liste d'annuaires, et copier des fichiers.

On peut lancer un transfert de fichier à partir d'une interface graphique ou taper directement une commande. Par exemple, dans un browser Web, un transfert sera initié avec une commande commençant par ftp://Le recours à ce protocole est particulièrement utile lorsque l'on développe une page sur une machine locale et que l'on veut la publier sur un site Web.

Fonctionnement

Le protocole FTP a besoin de deux canaux pour fonctionner

- Le canal de contrôle qui sert à envoyer les commandes comme le listing de dossier, le changement de dossier
- Le canal de données qui sert à envoyer les données au client

Le protocole FTP fonctionne suivant deux modes

Le mode passif

Le client se connecte sur le port 21 du serveur pour le transfert des données. Ce mode passe en général assez bien par les passerelles NAT.

Le mode actif

Le client se connecte depuis un port sur le port 21 du serveur pour le contrôle et le serveur se connecte depuis son port 20 sur le port du client pour envoyer les données. Ce mode ne peut pas passer par un NAT car il nécessite une connexion entrante vers la machine client.

III. 2.7. Le protocole SSH

SSH est un protocole réseau sécurisé permettant l'établissement de connexions interactives, l'exécution de commandes distantes, le transfert de fichiers et le relais d'applications TCP.

SSH met en jeu des mécanismes de chiffrement pour la confidentialité des données mais présente également des mécanismes d'authentification forte

Utilité de SSH

- ✓ Éviter la circulation en clair sur le réseau des mots de passe
- ✓ Renforcer l'authentification des machines
- ✓ Sécuriser le transfert de données

Fonctionnalités de SSH

Les fonctionnalités et garanties principales du protocole SSH sont :

confidentialité :SSH préserve la confidentialité des donnée en les chiffrant lorsqu'elles passent par le réseau.

Authentification : Le chiffrement des mots de passe ou des informations sensibles nécessaires pour l'authentification diminue le risque lié à l'espionnage du réseau.

L'intégrité :

L'autorisation :consiste à décider de ce quelqu'un peut faire ou non

Transfert : toutes ces données seront automatiquement chiffrées et leur intégrité sera contrôlée

III. 2.8. Le protocole Telnet

Telnet est un protocole simple de connexion à distance : il permet de transmettre des caractères entre une machine locale et une machine distante.

Il défini deux type (client, serveur)

- Le client Telnet
- Le serveur Telnet

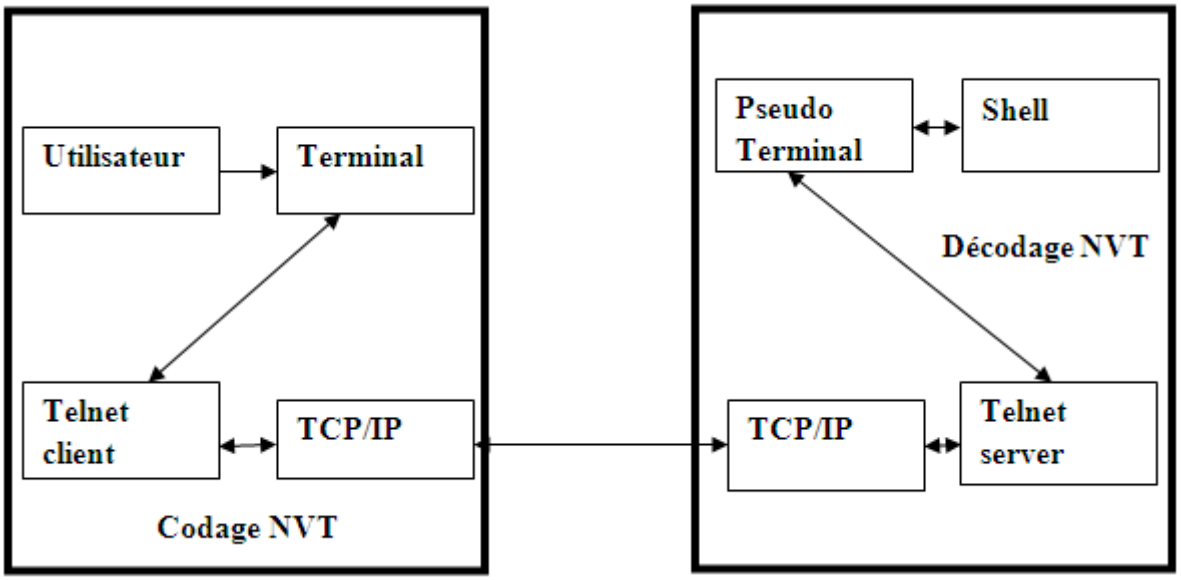


Figure 06 : fonctionnement de Telnet

III. 2.9. Le protocole SMTP

SMTP Simple Mail Transfer Protocol est un protocole standard permettant de transférer le courrier d'un serveur à un autre en connexion point à point. Il s'agit d'un protocole fonctionnant en mode connecté, encapsulé dans une trame TCP/IP.

Le protocole SMTP fonctionne grâce à des commandes textuelles envoyées au serveur SMTP

Pour pouvoir se connecter sur un serveur de courriers en mode SMTP où ESMTP, il suffit d'utiliser la commande texte HELO pour le protocole SMTP et EHLO pour le protocole ESMTP

Transfert

Il existe certains cas où l'information concernant un destinataire donnée dans la <forward-path > est incorrecte, mais le récepteur SMTP connaît la destination exacte. Dans un tel cas, l'une des réponses suivantes pourra être émise pour permettre à l'émetteur de contacter la bonne destination.

Vérification d'adresse et expansion de listes.

SMTP propose de fonctionnalités additionnelles, des commandes pour vérifier un nom de destinataire ou pour expanser une liste de diffusion. Ces deux opérations peuvent être menées respectivement avec les commandes VRFY et EXPN,

III. 2.10. Le protocole pop3

POP : Post Office Protocol est un protocole qui permet de récupérer les courriers électroniques situés sur un serveur de messagerie électronique. Ce protocole a été réalisé en plusieurs versions respectivement POP1, POP2 et POP3.

Commandes principales

DELE : numéro du message
LIST : donne une liste des messages ainsi que la taille de chaque message
RETR : numéro du message *récupère le message indiqué*
STAT : indique le nombre de messages
TOP : numéro du message nombre de lignes
Port 110

III. 2.11. Le protocole HTTP

HTTP : HyperText Transfert Protocol est un protocole de transfert de document hypertexte, Les documents hypertexte sont simplement les documents html des premières heures, en effet la première version du protocole était exclusivement réservée aux pages web.

La requête http : La requête utilisée dans cet exemple est la plus simple que l'on puisse trouver
La réponse http : Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur

III. 3. Les règles

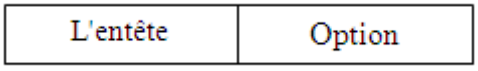
Apres que la conception des protocoles intervient à l'esprit on peut maintenant commencer à la notion des règles.

Comme les virus, la plupart d'activité d'intrus a une certaine sorte de signature. Les informations sur ces derniers sont employées pour créer des règles d'IDS.

Pour identifier les intrusions il y a une base de données pour les vulnérabilités connus que l'intrus veut exploiter, Ces attaques connues sont également utilisés comme signatures pour découvrir si quelqu'un essaye de les exploiter. Ces signatures peuvent être présentes dans la partie en-tête d'un paquet.

III. 3.1. Structure d'une règle:

La plupart des règles utilisées dans les IDS ont deux parties logique : en-tête de règle et options de règle comme montré sur la figure suivante :



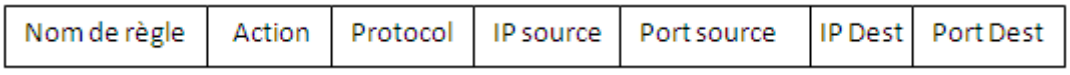
L'**en-tête** de règle contient des informations sur quelle mesure une règle prend. Il contient également des critères pour assortiment d'une règle contre des paquets de données.

La partie **options** contient habituellement un message d'alerte et une information au sujet dont une partie du paquet devrait être utilisé pour générer le message alerte.

Une règle peut détecter un ou plusieurs types d'activité d'intrusion. Les règles intelligentes devraient pouvoir s'appliquer aux signatures multiples d'intrusion.

III. 3.1.1 Les entêtes de règle

La structure générale d'un en-tête de règle d'IDS est montrée sur le figure suivante.



Le champ **nom de règle** concerne le type d'alerte affiché quand un intrus est détecté (eg :ICMP alerte !).

Le champ **action** détermine l'action qui sera prise quand les conditions de règle sont vérifiées.

Une mesure est prise seulement quand toutes les conditions mentionnées dans une règle sont vraies.

Le champ action peut prendre les valeurs suivantes :

- **alert** : générer une alerte + logger le paquet
- **log** : logger le paquet.
- **pass** : ignorer le paquet.
- **activate** : activer une règle dynamique.
- **dynamic** : définir une règle dynamique, qui est passive tant qu'elle n'est pas activée par une autre règle.
- **drop** : demander à iptables10 de bloquer le paquet, puis le logger.
- **reject** : demander à iptables de bloquer le paquet, puis le logger, et envoyer une commande TCP RST (reset) ou une réponse ICMP Host unreachable.
- **sdrop** : demander à iptables de bloquer le paquet. Ce dernier n'est pas loggé.

Le champ **protocole** spécifie le protocole pour lequel la règle s'applique. Les valeurs possibles sont : tcp, udp, icmp ou ip.

Les **adresses** peuvent être un hôte simple, des centres serveurs multiples ou des adresses de réseau. Notez qu'il y a deux zones adresses dans la règle. Les adresses de **source** et les adresses de **destination** sont déterminées selon leur classement dans la règle.

Le champ **ports** détermine les ports de source et de destination d'un paquet sur lequel la règle est appliquée. En cas de protocoles de couche réseau comme IP et ICMP, les numéros de ports n'ont aucune signification.

III. 3.1.2. Options de règles

Les options de règle forment le coeur du moteur de détection d'intrusion , combinant facilité d'utilisation, puissance et flexibilité. Toutes les options de règle sont séparées les unes des autres par un caractère point virgule ";". Les mots clés des options de règle sont séparés de leurs arguments avec un caractère deux points ":". Au moment de la rédaction, les mots clé d'option de règle disponibles sont :

- **msg** - affiche un message dans les alertes et journalise les paquets
- **ttl** - teste la valeur du champ TTL de l'entête IP
- **tos** - teste la valeur du champ TOS de l'entête IP
- **id** - teste le champ ID de fragment de l'entête IP pour une valeur spécifiée
- **fragbits** - teste les bits de fragmentation de l'entête IP
- **dsize** - teste la taille de la charge du paquet contre une valeur
- **flags** - teste les drapeaux TCP pour certaines valeurs
- **seq** - teste le champ TCP de numéro de séquence pour une valeur spécifique
- **ack** - teste le champ TCP d'acquittement pour une valeur spécifiée
- **itype** - teste le champ type ICMP contre une valeur spécifiée
- **icmp_id** - teste la champ ICMP ECHO ID contre une valeur spécifiée
- **content** - recherche un motif dans la charge d'un paquet
- **content-list** - recherche un ensemble de motifs dans la charge d'un paquet
- **offset** - modifie l'option content, fixe le décalage du début de la tentative de correspondance de motif
- **depth** - modifie l'option content, fixe la profondeur maximale de recherche pour la tentative de correspondance de motif
- **session** - affiche l'information de la couche applicative pour la session donnée
- **rpc** - regarde les services RPC pour des appels à des applications/procédures spécifiques

Msg

L'option de règle msg dit au moteur de journalisation et d'alerte le message à imprimer avec une sauvegarde du paquet ou une alerte.

Format : **msg: "<message texte>";**

TTL

Ce mot clé d'option est utilisé pour fixer la valeur time-to-live à tester. Le test qu'il effectué est réussi seulement sur une correspondance exacte. Ce mot de clé d'option était destiné à être utilisé pour détecter les tentatives de traceroute.

Format : **ttl: "<nombre>";**

TOS

Le mot clé "tos" vous permet de vérifier le champ TOS de l'entête IP pour une valeur spécifique. Le test effectué est réussi seulement sur une correspondance exacte.

Format : **tos: "<nombre>";**

ID

Ce mot clé d'option est utilisé pour tester une correspondance exacte dans le champ ID de fragment de l'entête IP. Quelques programmes de pirates (et d'autres programmes) fixent ce champ spécifiquement pour différents besoins, par exemple la valeur 31337 est très populaire avec certains pirates. Ceci peut être retourné contre eux en mettant en place une simple règle pour tester ceci et quelques autres "nombres de pirates".

Format : **id: "<nombre>";**

Fragbits

Cette règle inspecte les bits de fragment et le bit réservé dans l'entête IP. Il y a trois bits qui peuvent être vérifiés, le bit Reserved Bit (RB), le bit More Fragments (MF) et le bit Dont Fragment (DF). Ces bits peuvent être vérifiés pour une variété de combinaisons. Utilisez les valeurs suivantes pour indiquer les bits spécifiques : (R - Reserved Bit, D - DF bit, M - MF bit)

Format : **fragbits: <valeurs des bits>;**

Dsize

L'option dsize est utilisée pour tester la taille de la charge du paquet. Il peut être fixé à toute valeur, utilise en plus les signes supérieur/inférieur pour indiquer des intervalles et des limites. Par exemple, si vous savez qu'un certain service a un tampon d'une certaine taille, vous pouvez fixer cette option pour regarder les tentatives de débordement de tampons. Cela a l'avantage supplémentaire d'être une façon bien plus rapide de tester contre les débordements de tampons qu'une vérification de contenu de la charge.

Format : **dsize: [>|<] <nombre>;**

Content

Le mot clé content est une des fonctionnalités les plus importantes. Il autorise l'utilisateur de fixer des règles qui recherchent un contenu spécifique dans la charge du paquet et déclencher une réponse basée sur les données.

Format : **content: "<chaîne de contenu>";**

Offset

L'option de règle offset est utilisée comme un modificateur des règles utilisant le mot clé d'option content. Ce mot clé modifie la position de début de recherche pour la fonction de correspondance de motif depuis le début de la charge du paquet.

Ce mot clé d'option de règle ne peut pas être utilisé sans spécifier également l'option de règle content.

Format : **offset: <nombre>;**

Depth

Depth est une autre modification de l'option de règle content. Ceci fixe la profondeur maximale de recherche dans la fonction content de correspondance de motif de recherche depuis le début de sa région de recherche.

Format : **depth: <nombre>;**

Flags

Cette règle teste les drapeaux TCP pour une correspondance.

Format : **flags: <valeurs de drapeaux>;**

Seq

Cette option de règle se réfère aux numéros de séquence TCP. Essentiellement, il détecte si le paquet a un numéro de séquence statique fixé, et donc plutôt peu utilisé. Elle a été incluse pour le bien de l'exhaustivité.

Format : **seq: <nombre>;**

Ack

Le mot clé ack d'option de règle se réfère au champ d'acquittement de l'entête TCP. Cette règle a un propos pratique jusqu'ici : détecter les pings TCP NMAP.

Format : **ack: <nombre>;**

Itype

Cette règle teste la valeur du champ type ICMP. Il est fixé en utilisant la valeur numérique du champ.

Format : **itype: <nombre>;**

Icmp_id

L'option icmp_id examine le numéro ICMP ID d'un paquet ECHO ICMP pour une valeur spécifique. C'est utile car quelques programmes de canaux cachés utilisent des champs ICMP statiques quand ils communiquent.

Format : **icmp_id: <nombre>;**

Rpc

Cette option regarde les requêtes RPC et décode automatiquement l'application, la procédure et la version de programme, en indiquant un succès quand les trois variables correspondent toutes. Le format de l'option d'appel est "application, procédure, version". Les caractères génériques sont valides pour la procédure et les numéros de version et sont indiquées par une "*".

Format : **icmp_seq: <nombre,[nombre|*],[nombre|*]>;**

Content-list

Le mot clé content-list permet à de multiples chaînes de contenu d'être spécifiées à la place d'une unique option de contenu.

Format : **content-list: "<nom de fichier>";**

Exemple de règle

Nom de règle → règle de SMTP

Action de règle → alert

Protocole → icmp

IP source → 192.168.1.4

Port source → 25

IP destination → 192.168.1.1

Port destination → 25

Option de règle → msg: "email ID";

III 3.2. Organisation de règles selon l'action

Les types des règles peuvent être classés par catégorie dans trois types de base.

1- Les règles d'alerte.
2- Les règles Passage.
3- Les règles de log.

Si vous définissez vos propres types de règle, il faut vérifier leur séquence. Par exemple, si ont défini une règle de type **snmp_alerts**, le classement de l'application de règle sera:

Alert → Pass→ Log→snmp_alerts

Conclusion :

Plusieurs notions de base pour les IDS ont été présentés, et nous avons vu qu'il n'était pas aisé de créer une règle d'un IDS sans connaissance de protocoles.

Sommaire

Chapitre IV
implementation

Sample output to test PDF Combine only

Logiciel et outil utilisés

Les outils de développement que nous avons utilisés sont:

Java

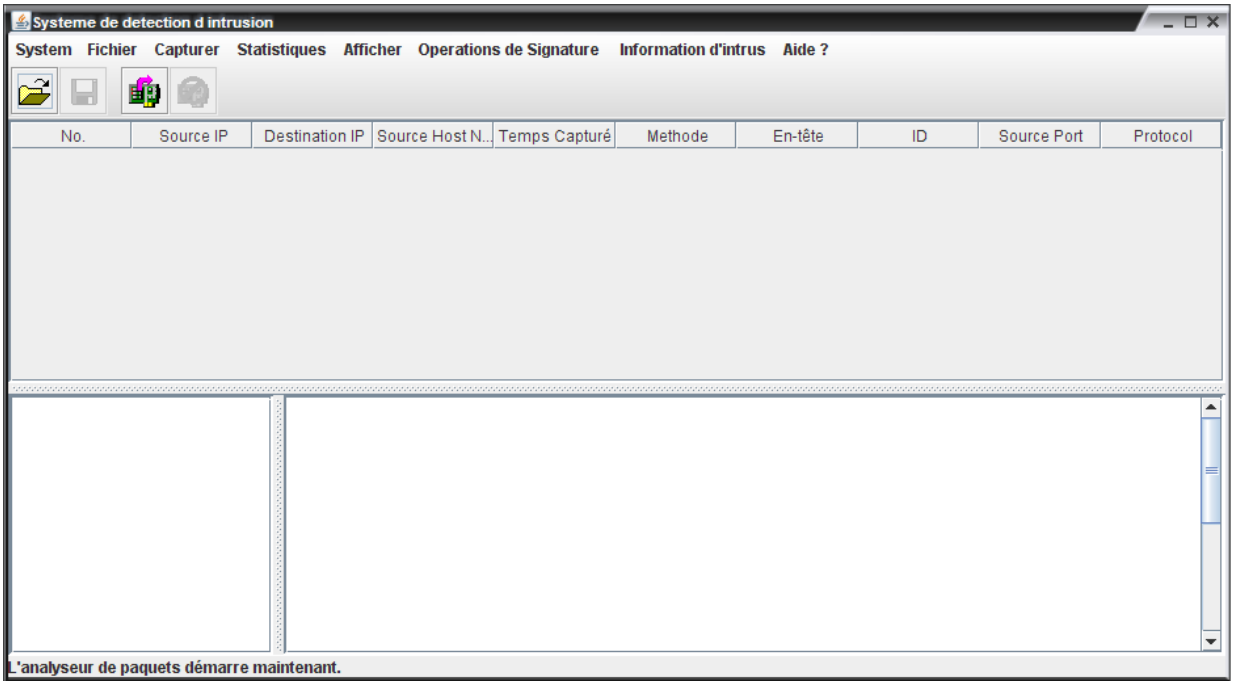
Java est un langage de programmation développé par Sun Microsystems en 1995 comme composant de noyau de la plateforme de Java sun. son syntaxe dérive beaucoup de C et C++ mais a un modèle plus simple d'objet et peu d'équipements de bas niveau.
Des applications de Java sont typiquement compilées au bytecode qui peut fonctionner sur n'importe quel Java virtuel machine (JVM) indépendamment d'architecture d'ordinateur.

JCreator :

JCreator est un IDE puissant pour des technologies de Java™

Jpcap

Jpcap est une bibliothèque open source pour capturer et envoyer des paquets de réseau pour les applications Java.
. Il fournit des équipements à:
•capturer les paquets en temps réel.
•permet de sauvegarder les paquets capturés dans un fichier et la lecture hors connexion.
•identifie automatiquement les types de paquet et produise object de Java correspondant (pour Ethernet, paquets IPv4, Ipv6, ARP/rarp, TCP, UDP, et ICMPv4).
•filtrez les paquets selon des règles personnalisées par l'utilisateur.
•envoyez les paquets au réseau
Jpcap est basé sur libpcap/winpcap, et est mis en application dans C et Java.



.figure 01 : menu principale

1- **système** : contient les sous menus (nouvelle fenêtre et fermer).
2- **Fichier** : contient les sous menus (ouvrier et enregistre), permettant le sauvegarde des paquets capturé et leurs lecture hors connexion.
3- **Capturer** : permet le démarrage et l'arrêt de capture.
4- **Statistique :** contient deux sous menus pour afficher les Statistiques continues et cumulatives para port les déférents protocoles de couche (application, liaison, transport,) aussi que les informations globale et la mémoire libre.
5- **Afficher** : pour affiche les éléments concernant la premier partie de chapitre 3
6- **Operations de signature** : contient deux sous menus pour l'ajout et l'affichage des regles.
7- **Information d'intrus :** affiche la liste noire des intrusions.
8- **Le tableau de trafic :** affiche les paquets circulant dans le réseau**.**
9- **Espace d'exploration :** affiche les informations globales pour un paquet spécifique**.**
10- affiche les informations sur le contenue d'un paquet spécifique**.**

Ajout d'une règle

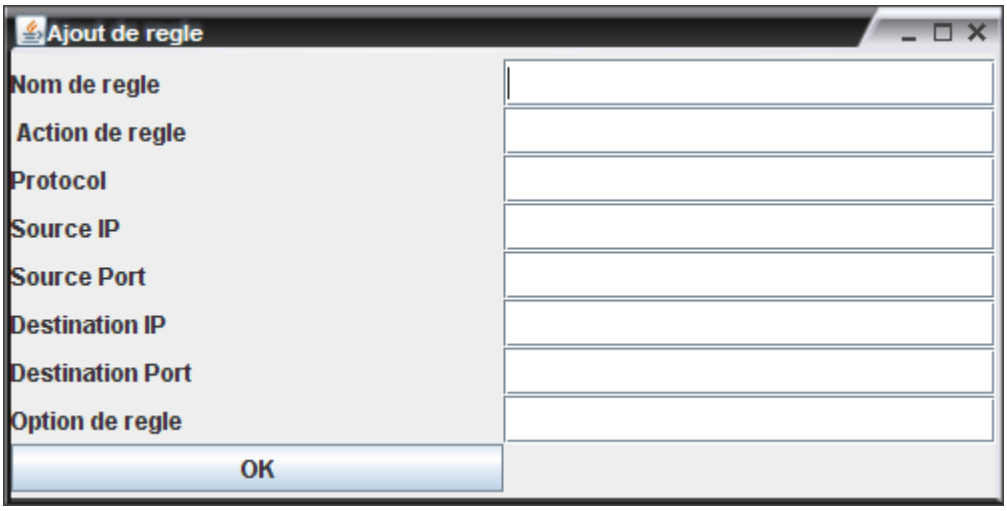


Figure 02 : Ajout d'une règle

Affichage de tous les regles

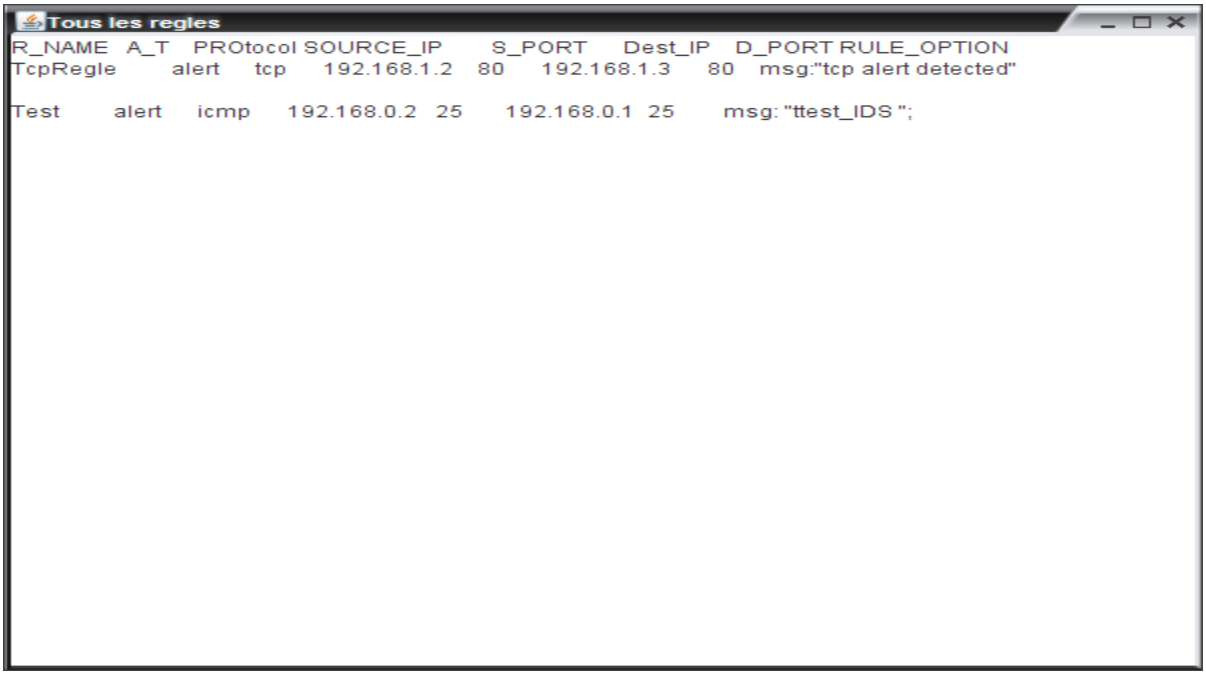
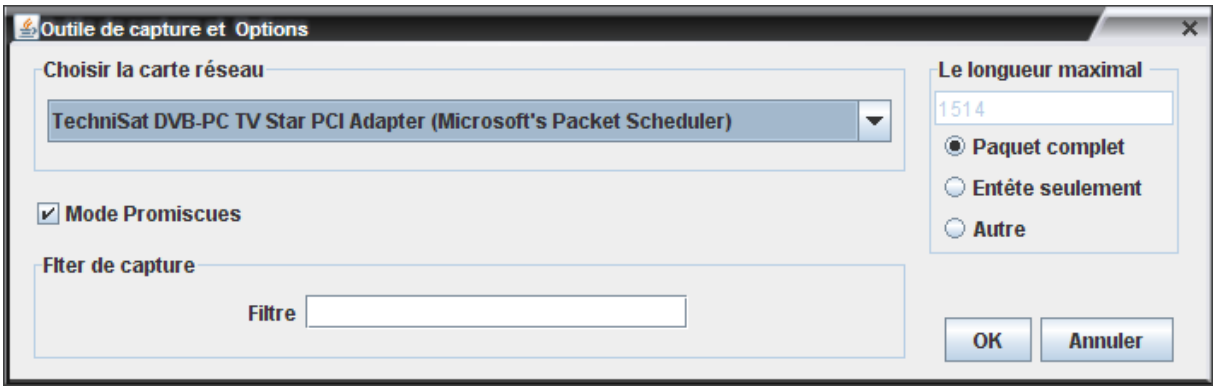


Figure 03 : Affichage de tous les regles

Démarrage de capture



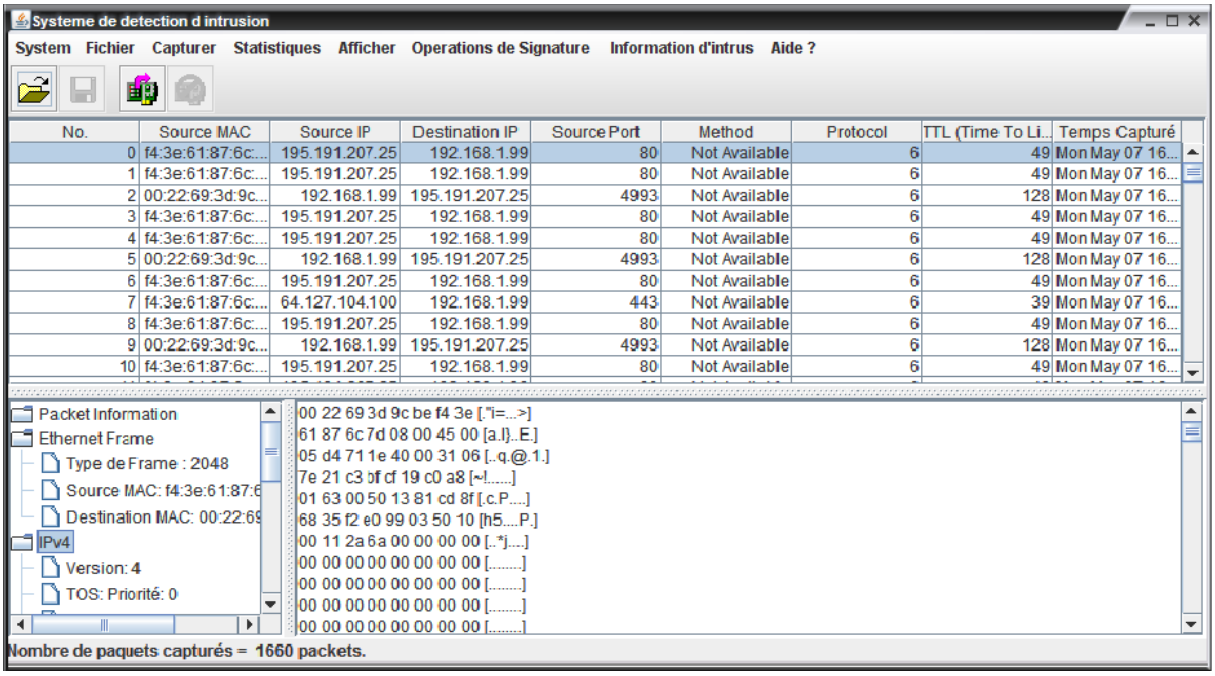


Figure 04 : Démarrage de capture

Les statistiques cumulatives pour la proportion du protocole de couche transport.

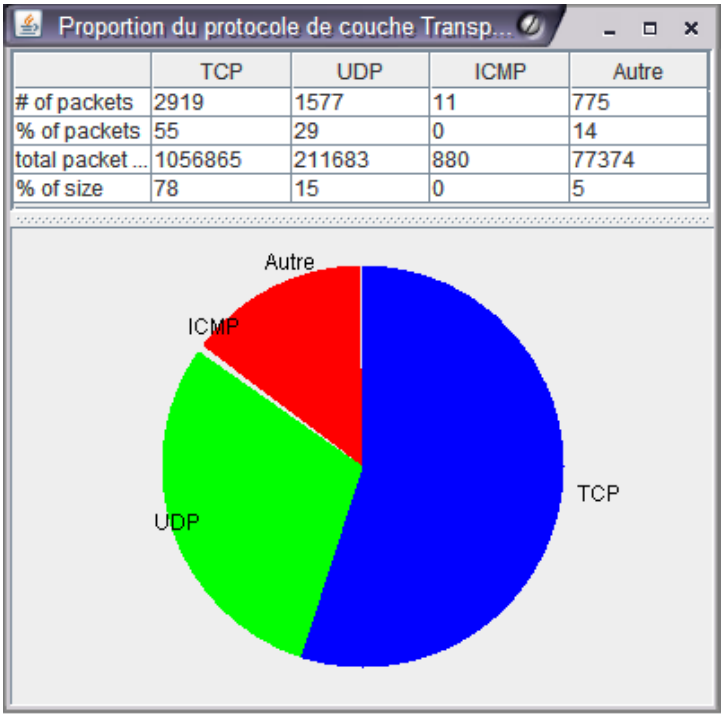


Figure 05 :Les statistiques cumulatives

Les statistiques continues pour la proportion du protocole de couche transport.

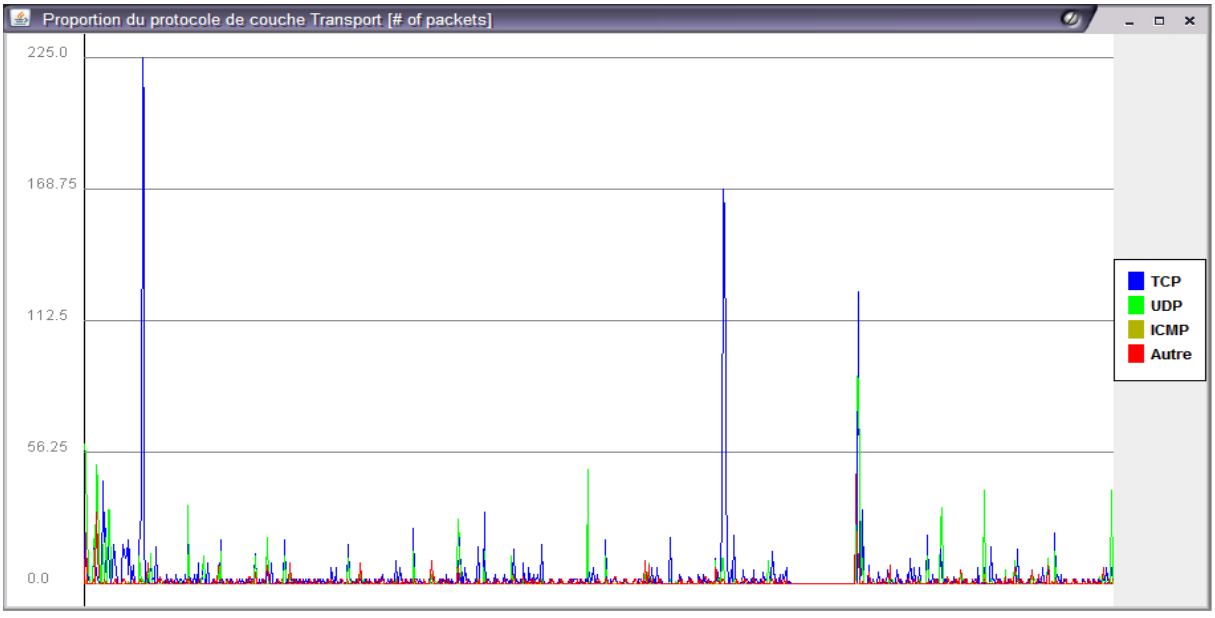


Figure 06 :Les statistiques continues

Génération d'alerte.

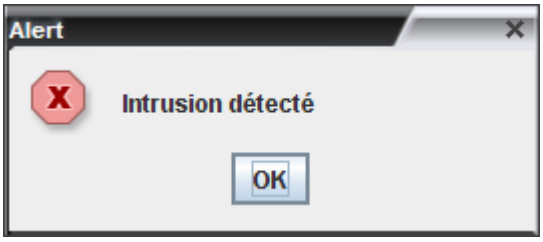


Figure 07: Génération d'alerte.

Liste des intrusions : représente la liste des intrusions détectées

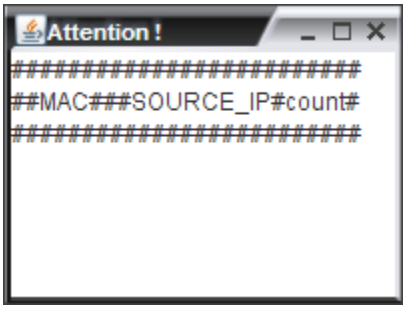


Figure 08: Liste des intrusions

Quelques codes source essentiel utilisées :

Ce code est pour detecter d'abord les carte réseaux installés puis les utilisés pour le capture de paquet.

```
public void capturePacketsFromDevice() {
      if(jpcap!=null)
         jpcap.close();
      jpcap = JDCaptureDialog.getJpcap(frame);
      clear();

      if (jpcap != null) {
         isLiveCapture = true;
         frame.disableCapture();

         startCaptureThread();
```

```
    }
  }
```

Les analyseurs de protocoles

Exemple le protocole TCP

```
public TCPAnalyzer() {
   this.layer = TRANSPORT_LAYER;
 }

 public boolean isAnalyzable(Packet p) {
  return p instanceof TCPPacket;
 }

 public String getProtocolName() {
  return "TCP";
 }

 public String[] getValueNames() {
  return valueNames;
 }

 public void analyze(Packet p) {
  this.values.clear();
  if (!isAnalyzable(p)) return;
  TCPPacket tcp = (TCPPacket)p;
  this.values.put(valueNames[0], new Integer(tcp.src_port));
  this.values.put(valueNames[1], new Integer(tcp.dst_port));
  this.values.put(valueNames[2], new Long(tcp.sequence));
  this.values.put(valueNames[3], new Long(tcp.ack_num));
  this.values.put(valueNames[4], new Boolean(tcp.urg));
  this.values.put(valueNames[5], new Boolean(tcp.ack));
  this.values.put(valueNames[6], new Boolean(tcp.psh));
  this.values.put(valueNames[7], new Boolean(tcp.rst));
  this.values.put(valueNames[8], new Boolean(tcp.syn));
  this.values.put(valueNames[9], new Boolean(tcp.fin));
  this.values.put(valueNames[10], new Integer(tcp.window));
 }

 public Object getValue(String valueName) {
  return this.values.get(valueName);
 }

 Object getValueAt(int index) {
  if ((index < 0) || (index >= valueNames.length)) return null;
  return this.values.get(valueNames[index]);
 }

 public Object[] getValues() {
  Object[] v = new Object[valueNames.length];

  for (int i = 0; i < valueNames.length; i++) {
```

```
      v[i] = this.values.get(valueNames[i]);
    }
    return v;
  }
}
```

Conclusion general

Bien que répandus dans les organisations aujourd'hui, les systèmes de détection d'intrusions ne représentent qu'un maillon d'une politique de sécurité. En effet, même si ceux-ci permettent la détection, parfois l'arrêt, des intrusions, ils restent néanmoins vulnérables eux aussi faces aux attaques externes.

C'est pourquoi, pour une sécurité optimale, ces outils doivent être couplés à d'autres, comme l'indispensable pare-feu. Mais ils doivent aussi être mis à jour, aussi bien le cœur du logiciel comme la base de signatures, qui constitue la base d'une détection efficace. Il faut également coupler les systèmes de détection entre eux : c'est-à-dire ne pas hésiter à placer des NIDS, HIDS et KIDS dans le même réseau. Leurs rôles sont différents, et chacun apporte ses fonctionnalités.

Toutefois, et nous terminerons par ceci, même si une certaine maturité dans ce domaine commence à se sentir, le plus important reste de savoir de quoi il faut se protéger. Les failles les plus répandues proviennent généralement de l'intérieur de l'entreprise, et non de l'extérieur. Des mots de passe simples, des droits d'accès trop élevés, des services mal configurés, ou encore des failles dans les logiciels restent la bête noire en matière de sécurité.

[1] : Bernard Cousin ***Sécurité des réseaux informatiques***

[2] : Eric Maiwald, ***Sécurité des réseaux***

[3] : Randal Vaughn et Gadi Evron ***DNS Amplification Attacks***

[4] : Laurent LEVIER ***Attaque des réseaux***

[5] : Sean Whalen ***An Introduction to ARP Spoofing***

[6] : Stéphane Gill ***Type d'attaques***

[7] : Vamshidhar Chillamcharla ***ARP Spoofing***

[8] : Laurent Joncheray ***Simple Active Attack Against TCP***

[9] : Victor Velsco ***Introduction to IP Spoofing***

[10] : ADM Crew ***DNS ID Hacking***

[11] D.E. Denning ***An Intrusion-Detection Model***

[12] K. Ilgun, R.A. Kemmerer, P.A. Porras **State transition analysis: a rule-based intrusion detection approach**

[13] K. Müller ***IDS - Systèmes de Détection d'Intrusion, Partie I***

[14] K. Müller ***IDS - Systèmes de Détection d'Intrusion, Partie II***

[15] David Burgermeister, Jonathan Krier ***Systèmes de Détection d'Intrusion***

[16] Servin Claude **Réseaux et Télécoms**

[17] Douglas Comer **TCP/IP - Architecture, protocoles**

[18] Cyril pain barre **Format des messages ICMP**

[19] François laissus **Cours d'introduction à TCP/IP**

[20] RFC 826 **Ethernet Address Resolution Protocol**

[21] RFC 793 **Transmission Control Protocol**

[22] RFC 792 **Internet Control Message Protocol**

[23**]** RFC4253 **Secure Shell**

[24] RFC 2616 **HTTP**