الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

UNIVERSITE SAAD DAHLEB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT INFORMATIQUE

MEMOIRE DE FIN D'ETUDE
POUR L'OBTENTION DU DIPLOME D'INGENIEUR D'ETAT EN INFORMATIQUE

TITRE :

CONCEPTION ET REALISATION D'UN SCANNER DE SECURITE

Présenté par :
Mr. BEN CHIBANE Mohamed.
Mr. BRAHMI Abdelkader.

Proposé et dirigé par :
Mr. OULED BRAHAM Malek

2002/2003

LISTE DE TABLEAUX

Remerciement

Nous exprimons nos plus vifs remerciements à notre promoteur Monsieur **Malek OULD BRAHAM**, de nous avoir pris en charge tout au long du projet et nous lui exprimons notre gratitude pour ses conseils constructifs et ses encouragements.

Nous tenons à remercier Mdm Benstiti et Mlle Boustia, de nous avoir suivies pour terminer notre travaille.

Nous tenons à remercier les enseignant de département d'informatique de l'université de Blida pour avoir contribué à notre formation.

Nous exprimons notre profonde reconnaissance à nos parents et tous ceux qui nous ont aidé, de prés ou de loin, matériellement ou moralement, et partager nos peines pour voir ce modeste ouvrage naître,et à notre collègue kedjour hichem .

Dédicaces

Le présent mémoire est dédié à :

Mon dieu.

Mes très chers parents.

Mes très chers frères et mes très chères sœurs,

Mes oncles, mes tantes et leurs enfants qui sont pour moi comme des frères.

A toute ma famille.

A tous mes amis,

A tous ceux qui me sont chers, et dont je n'ai guère pu citer les noms,

BENCHIBANE MOHAMED

Dédicace

Le présent mémoire est dédié à :

Mon dieu.

Mes très chers parents.

Mes très chers frères et mes très chères sœurs,

Mes oncles, mes tantes et leurs enfants qui sont pour moi comme des frères.

A toute ma famille.

A tous mes amis, président, hichem, anis

A tous ceux qui me sont chers, et dont je n'ai guère pu citer les noms,

BRAHMI abdelkader

RÉSUME

Le présent travail consiste à la conception et la réalisation d'un scanner de sécurité pour les réseaux (c'est un –IDS- système de détection d'intrusions).

L'objectif fondamental de ce système est de déceler les failles de sécurité les plus fréquentes dans les systèmes connectés au réseau.
L'outil comporte aussi une base de données de failles, cette base n'est rien d'autre qu'un ensemble des signatures d'attaques, où la présence d'une signature dans les paquets qui circulent dans le réseau signifier une attaque.
C'est un scanner ouvert ; car il permet l'enrichissement de sa base de signature pour déceler de nouvelles failles et l'ajout de nouvelles stratégies de sécurité, ce scanner est implémenté autour des plateformes Windows.
En fin, le système offre une interface graphique qui permet : de faire une vérification interactif, et émettra dans le cas échéant une alerte SNMP ou Syslog vers l'administrateur de réseau.

ABSTRACT

The present works consist to the conception and realisation of scanner of security for network (it's an –IDS- intrusions detection system).

The objective fundamental of this system is to reveal the most fails of security in the networks systems.
This tool includes a data base of the fails, this base it's a set of attacks signatures, where the presences of any signature in the packs whose circulate in the network signify an attack.
This is an open source scanner; because it allow the enrichment of it's signature base to disclose a new fail and to make additions of a new strategy of security, this scanner is implemented around the Windows platform's.
At last, the system gives a graphic interface which allows: to do an interactive verification, and send in the falling case an SNMP or Syslog alert to the administrator of the system.

CHAPITRE I

SECURITE INFORMATIQUE

INTRODUCTION GÉNÉRALE

Avec le poids économique de plus en plus important de l'informatique dans l'industrie, et plus encore du fait de la dépendance grandissante envers les réseaux de communication, et étant donnée le nombre de plus grand de réseaux interconnectés, la sécurité des données informatique est aujourd'hui un problème crucial. Ce problème est aggravé avec l'existence des attaques en particulier le risque d'intrusion informatique.

C'est la raison pour laquelle que les institutions et les entreprises à l'exemple du Microtel ne doivent recourir à l'utilisation des technologies de l'information et la communication et surtout Internet, sans avoir au préalable analyser les risques et les menaces et s'être prémuni contre ces dernières.
Cette situation est déjà vécue au niveau du Microtel. En particulier dans l'installation de réseau pour l'université de Blida.
C'est pour cela que Microtel se doit avant tout de se prémunir de ces attaques grâce à la mise en place d'un dispositif de sécurité informatique tel qu'un scanner de sécurité.

Notre projet consiste à « la mise en œuvre d'un scanner de sécurité » qui permettra de déceler les failles de sécurité les plus fréquentes dans les systèmes connectés au réseau. Ces failles seront définies dans une base de données. Le scanner comparera les résultats obtenus à partir d'une station à la base de données et émettra dans le cas échéant une alerte SNMP vers une station d'administration.

Aborder la sécurité des systèmes d'information est souvent considéré comme une tâche technique difficile généralement réservée à des spécialistes.
Pour cette raison on a suive les étapes suivante :
Nous présenterons dans le premier chapitre les différents services qu'un système de sécurisation se doit de prendre en compte, à savoir la disponibilité

de service, l'authentification, l'intégrité et la confidentialité. Ces différents aspects seront décrites en détails en appuierons sur les formes de sécurité et les menaces sur elle.

Dans le chapitre suivant nous aborderons les systèmes informatiques interconnectés. Dans cette partie nous étudierons avec détails les architectures OSI et TCP/IP, fur et mesure on recensera les failles de sécurité des protocoles, et en fin on évaluera les failles de sécurité des systèmes Windows.

A la suite de ce chapitre, on abordera les solutions de la sécurité à savoir la prise en compte des services de sécurité de manière unifiée. Nous présenterons les différents outils de sécurité : les VPN, les firewall et enfin les IDS, nous insistons particulièrement sur les IDS.

Le dernier chapitre sera consacré à la conception et la réalisation de notre scanner de sécurité, nous basons sur le développement de notre solution, cette partie traite les techniques de détection d'intrusion que nous avons retenu pour notre projet, qui traite les signatures des attaques, en utilisons l'approche pattern matching.

1. INTRODUCTION

Dans un domaine aussi complexe et vaste que la sécurité, la première de nos préoccupations a été de fixer les concepts et les terminologies. C'est pourquoi nous débutons cet ouvrage en présentant les services qu'un système de sécurisation peut rendre, et les principaux services y répondant.

Dans ce chapitre, nous étudions la sécurité informatique et en particulier la sécurité dans les réseaux. D'abord, nous donnons une introduction très générale, puis nous étudions ses objectifs et ses menaces. Enfin, nous voyons plus en détail les mécanismes possibles pour assurer la sécurité et l'évaluation de la sécurité d'un système.

1.1 TERMINOLOGIE DE LA SECURITE

Le terme « sécurité » est utilisé dans le sens de minimiser les vulnérabilités d'actifs et de ressources.

-Un actif est tout élément de valeur.

-Une vulnérabilité est toute faiblesse qui pourrait être exploitée pour violer un système ou les informations qu'il contient.

-Une menace est violation potentielle de la sécurité [ISO 89].

La définition la plus générale de la **sécurité informatique** est d'assurer le bon fonctionnement d'un système informatique. Mais une définition plus précise serait de définir la sécurité comme étant la capacité d'un système à protéger des objets en respectant la confidentialité, l'intégrité, la disponibilité de service et l'authentification [OLO 92].

Un **objet** d'un système informatique est une entité passive qui contient ou reçoit de l'information. (CPU, disque ou programme…) L'accès à un objet implique l'accès aux informations qu'il possède. [DOS 83].

Un **sujet** d'un système informatique est un entité active, généralement sous la forme d'une personne, d'un processus ou d'un périphérique qui produit un flux d'information entre les objets ou qui change l'état du système. [DOS 83].

Les mots ***hackers*** ou ***crackers*** se réfèrent à ceux qui violent la sécurité en s'introduisant par effraction dans les systèmes, et à n'importe quelle personne effectuant des recherches sur l'intégrité et la sécurité d'un système. En général, il s'agit de programmeurs. Ils possèdent des connaissances tant sur le plan matériel que logiciel, et sont capable de pénétrer des systèmes en mettant en œuvre des pratiques innovatrices.

La définition donnée ci-dessus de la sécurité est similaire à d'autres définitions, mais au lieu de définir la sécurité comme étant la protection de l'information seulement,

cette définition inclus la protection des ressources du système tel que la protection contre l'utilisation illégale du temps CPU,

2. LES FORMES DE LA SECURITE

Un attaquant peut trouver différent objet à attaquer dans un système dans le but d'obtenir une information spécifique : les composants logiciels d'un système peuvent être attaqués, l'installation physiques des ordinateurs peut être attaquer …, Donc il est nécessaire de diviser la sécurité en différentes formes qui rassembleront les caractéristiques de la sécurité semblables ensemble.

Cette structure est basée sur les emplacements où on peut trouver des vulnérabilités du système : vulnérabilité dans le matériel, vulnérabilité dans l'information ou le logiciel, et vulnérabilité dans l'organisation administrative du système, comme le montre la figure 1.1.

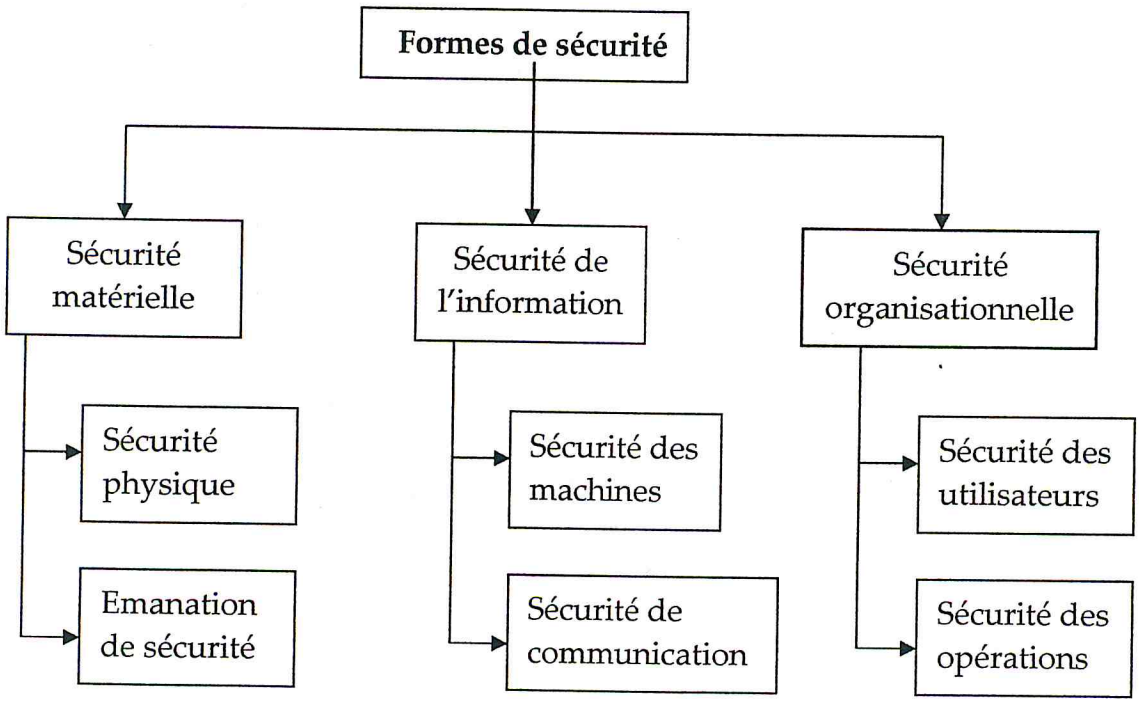


Figure 1. 1: Les formes de la sécurité

2.1 LA SECURITE MATERIELLE

La sécurité matérielle traite la protection des objets des vulnérabilités présentes lors de la manipulation du matériel. Elle peut être divisée en deux parties :

Sécurité physique.

Sécurité d'émanation.

2.1.1 SECURITE PHYSIQUE

Concerne la protection du matériel du système des menaces physiques externes, tel que les intempéries, le vol, les tremblements de terre, les inondations…, tous les équipements contenant des informations sensibles doivent être protégés.
Ce problème peut être résolu en plaçant l'équipement dans un environnement assez sécurisé pour contenir les informations traitées par cet équipement ; la sécurité physique traite la manière de créer et maintenir un tel environnement (voir Annexe A).

2.1.2 SECURITE D'EMANATION

Traite la protection de l'émission d'information par le matériel du système, par exemple émission électromagnétique ou émission audio.

2.2 SECURITE DE L'INFORMATION

La sécurité de l'information est la protection des objets des vulnérabilités présentes dans l'architecture du système, c'est à dire vulnérabilités dans le logiciel ou le matériel et dans la combinaison des deux, elle peut être divisée en :

2.2.1 SECURITE DES MACHINES

Concerne la protection des objets contre les expositions et les attaques qui font usage des vulnérabilités dans l'architecture du système. Elle traite d'une large variété de problèmes : quel mécanisme de chiffrement doit être choisi ? , Comment le mécanisme de contrôle d'accès doit-il fonctionner ?...

2.2.2 SECURITE DE COMMUNICATION

Concerne la protection de l'information durant son transport. Lors du transport d'objets, entre les machines, une attaque peut être actionne dans le but d'interagir avec le processus de communication. Par exemple, pour modifier, retransmettre, réordonner ou détruire l'information. Aussi les objets doivent être protégés contre l'exposition durant la transmission.

2.3 SECURITE ORGANISATIONNELLE

La sécurité organisationnelle est la protection des objets contre les vulnérabilités causées par les utilisateurs et les menaces contre l'organisation de la sécurité. Elle peut être divisée en deux parties : la sécurité des utilisateurs et la sécurité des opérations.

2.3.1 SECURITE DES UTILISATEURS

La sécurité des utilisateurs est la protection des objets contre les attaques des utilisateurs légitimes. Les utilisateurs d'un système ont accès à différent objets, d'où la

nécessité de mécanismes de protection contre les utilisateurs qui abusent de leurs privilèges. Les raisons qui poussent un utilisateur a abuser de ses privilèges sont nombreuses : il peut y avoir un intérêt personnel tel que l'argent, ou une vengeance contre la compagnie ...etc. La sécurité des utilisateurs peut inclure les expositions émanant des utilisateurs légitimes.

En général, les utilisateurs autorisés constituent une plus grande menace, que les attaquants externes. Les statistiques montrent que seulement 10% des crimes de piratage informatique sont causés par des attaques externes, 40% par des attaques de l'intérieur et 50% par d'anciens employés comme acte de vengeance [IRW 99]. En clair, la sécurité des utilisateurs doit avoir une grande influence sur les mécanismes de sécurité à implémenter dans le système (voire Annexe A).

2.3.2 SECURITE DES OPERATIONS

La sécurité des opérations concerne la protection des objets contre les vulnérabilités présentes dans l'organisation qui maintient la sécurité du système. La sécurité des opérations régularise la façon dont toutes les autres formes de sécurité doivent être implémentées et comment le système devrait être exploité. Elle traite des moyens à mettre en œuvre pour renforcer les règles de sécurité établies dans la politique de sécurité voire (Annexe A), les actions à prendre quand une violation de la sécurité est détectée, ...etc. Il est important que les personnes responsables du maintien de la sécurité du système soient continuellement à jour, pour le renforcement des mécanismes de sécurité.

Notant que la sécurité de fonctionnement telle que définie ci-dessus ne peut pas être la seule cible d'attaques. Par exemple un attaquant peut exploiter une vulnérabilité dans un logiciel qui n'est pas à jour, mais cette attaque peut être dirigée contre une autre forme de sécurité, dans ce cas il s'agit de la sécurité des machines. Il est important de faire cette distinction sinon toutes les attaques seraient vue comme étant dirigées contre la sécurité de fonctionnement puisque cette dernière maintient la totalité de la sécurité du système. En d'autres mots, une vulnérabilité (une défaillance) dans la sécurité de fonctionnement peut engendrer une attaque contre une autre forme de sécurité.

3. MENACES ET TYPES D'ATTAQUES

Les menaces peuvent être vues comme une potentielle violation de la sécurité, elles existent à cause des vulnérabilités et faiblesses d'un système, elles comprennent les éléments suivants : [ISO 89]

Destruction d'information ou d'autres ressources.

Corruption ou modification d'informations.

Vol, suppression ou perte d'informations ou d'autres ressources.

Divulgation d'informations.

Interruption de services.

Les menaces peuvent être classées en menaces accidentelles ou menaces intentionnelles (attaques) et elles peuvent être actives ou passives.

3.1 LES MENACES ACCIDENTELLES

Les menaces accidentelles sont celles qui existent sans qu'il y ait préméditation. Des exemples des menaces accidentelles qui se sont concrétisées sont : défaillance de système, bévues opérationnelles et bogue de logiciels [ISO 89]. Un exemple concret de ce type de menaces serait l'envoi par un utilisateur d'un mail confidentiel à la mauvaise personne par erreur.

3.2 LES MENACES INTENTIONNELLES

Les menaces intentionnelles peuvent aller de l'examen fortuit, utilisant des outils de contrôle facilement disponibles, aux attaques sophistiquées, utilisant une connaissance spéciale du système. Une menace intentionnelle qui se concrétise peut être vue comme une «attaque ». [ISO 89]

Une attaque est une action exécutée par une entité avec l'intention de violer la sécurité. Exemple d'attaques : la destruction, modification, fabrication, interruption et interception de données. Le résultat d'une attaque pourrait être la violation de la confidentialité, de l'intégrité, de la disponibilité d'un objet, ou la modification d'un objet [OLO 92].

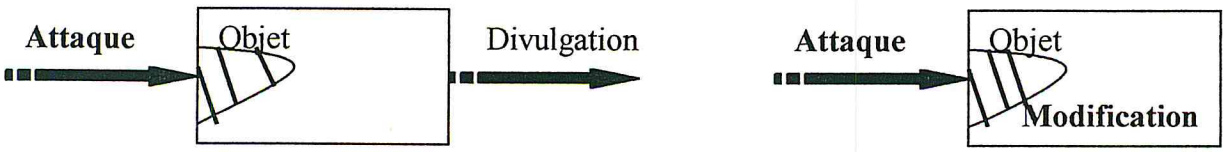


Figure 1. 2: attaque contre un objet

Une attaque peut être **directe** ou **indirecte** :

Une attaque directe vise directement un objet, plusieurs composants du système peuvent être attaqués avant d'accéder à l'objet final. Dans ce cas tous les objets intermédiaires sont la cible d'attaques directes.

Dans une attaque indirecte, l'information est reçue d'un objet sans attaquer l'objet soit même [OLO 92].

Il existe deux types d'attaques : les attaques passives et les attaques actives.

3.2.1 LES ATTAQUES PASSIVES

Les attaques passives sont celles qui, si elles se concrétisent, ne produiraient aucune modification d'informations contenues dans le(s) système(s) et avec lesquelles ni le fonctionnement ni l'état du système ne changent [ISO 89].
On peut en donner les exemples suivants : [VER 95]

- Ecoute physique (branchement sur une ligne de transmission, capture des signaux hertziens, ...)
- Analyse de trafic, afin de découvrir l'existence d'une communication entre deux sites. (Utilisation de ***SNIFFER***).
- Déduction par inférence, visant à déduire des informations confidentielles à partir de données publiques
- Utilisation de canaux cachés. Cette méthode consiste à utiliser des brèches des protocoles de protection d'un système.

En général, il est très difficile de détecter les attaques passives puisqu'elles ne perturbent pas et elles n'interagissent pas avec le fonctionnement normal du système [OLO 92].

3.2.2 ATTAQUES ACTIVES

Les attaques actives envers un système comprennent l'altération d'informations contenues dans ce système, ou modification de l'état ou du fonctionnement du système. Une modification malveillante des tables de routage d'un système par un utilisateur non autorisé est un exemple de menace active [ISO 89].
Les attaques actives contrairement aux attaques passives sont plus faciles à détecter si les précautions appropriées sont prisent [OLO 92].

3.3 QUELQUES TYPES D'ATTAQUES SPECIFIQUES

Les paragraphes suivants passent brièvement en revue quelques-unes des attaques particulièrement intéressantes dans un environnement de traitement de données ou de communication de données.

3.3.1 DEGUISEMENT (MASCARADE)

Le déguisement est le procédé par lequel une entité se fait passer pour une autre. Le déguisement est généralement utilisé avec d'autres formes d'attaques actives, surtout le fait de rejouer (*reply*) et la modification des messages. Par exemple dans la figure 1.3, des séquences d'authentification peuvent être capturées et rejouées après qu'une séquence d'authentification correcte ait lieu. [ISO 89].

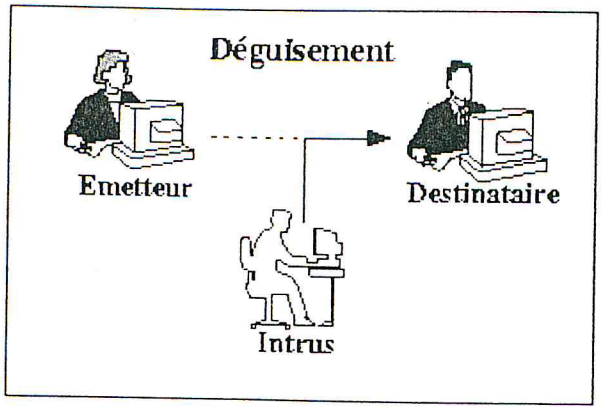


Figure 1. 3: Le déguisement.

3.3.2 LE DENI DE SERVICE (DENIAL OF SERVICE DOS)

Le déni de service a lieu lorsqu'une entité ne remplie pas sa propre fonction ou agit de façon qui empêche d'autres entités de remplir leurs fonctions ou d'utiliser le système [ISO 89].

Un déni de service traditionnel est réalisé en exploitant un débordement de tampon (buffer overflow), en épuisant les ressources système ou en exploitant un bug dans le système qui résulte en un système non fonctionnel (par exemple l'attaque par SYN flooding).

A l'été 1999, une nouvelle espèce d'attaque à été développée appelée «attaque par déni de service distribué » *Distributed Denial of Service (DDoS) attack* en visant plusieurs sites d'universités et sites commerciaux. Ce type d'attaque utilise une multitude de machine opérant en concert pour attaquer un réseau ou un site [CRI 00].

3.3.3 CHEVAL DE TROIE (TORJAN HORSE)

Un cheval de Troie est un Programme informatique qui a une fonction utile réelle ou apparente, mais contient des fonctions cachées qui exploitent furtivement les autorisations légitimes du processus d'appel, au détriment de la sécurité [LAR 96]. Un exemple d'un cheval de Troie est un éditeur de texte qui fait des recherches de mots clés spécifiques, et si un mot clé est trouvé une copie du document est envoyée à une autre personne. [OLO 92]

3.3.4 LES VIRUS ET LES VERS (VIRUSES AND WORMS)

Les virus et les vers sont des programmes ou des séquences de code conçu pour diffuser des copies d'eux même à d'autres programmes ou à d'autres ordinateurs. En infectant des programmes de cette façon, un virus se propagera soit même dans le système. La fonction d'un virus ou d'un ver est de perturber les services d'un système ou d'installer un cheval de Troie [OLO 92].

Un système bien conçu est comme une porte d'un immeuble, si la porte et la clé sont assez bien conçues, la plupart des intrus n'auront pas accès à l'immeuble. Mais aussi si la clé n'est pas assez maniable et facile à utiliser, il y aura une tendance à ne pas l'utiliser pour sa fonction spécifique, au moins pour une courte période de temps, et de cette façon créer un terrain favorable à l'intrus. La même règle est valable pour la politique de sécurité : les outils utilisés pour le renforcement de la sécurité doivent être assez bons et en même temps assez faciles a utiliser, pour qu'ils soient utilisés et acceptés par les utilisateurs du système [OLO 92].
Une action, intentionnelle ou non, qui viole les règles établies dans la politique de sécurité est une violation de sécurité.

5. LES MESURES DE SECURITE

Pour assurer la protection du réseau et de l'information qui y enregistrée ou qui y circule, on peut utiliser des services, des mécanismes ou des procédures que l'on nomme, de façon générale, des solutions ou des mesures de sécurité. Les mesures de sécurité consistent en un ensemble de mécanismes, des procédures et d'autres moyens qui sont mise en œuvre sur le réseau, afin de réduire les risques auxquels celui-ci est exposé. Par exemple, un service d'authentification aide à réduire le risque que des personnes non autorisées aient accès au réseau. Il importe de noter que les mesures de sécurité ne devraient pas être mises en place tant que l'on n'aura pas défini et au préalable une politique complète en matière de sécurité du réseau. En fait, les solutions et mesures de sécurité du réseau doivent être choisis afin de régler les problèmes indiqués dans la politique de sécurité réseau.
Dans la présente section, nous traitons des mesures suivantes :

- Confidentialité des données et de l'information ;
- Authentification ;
- Intégrité du système et des données ;
- Disponibilité.

5.1 SERVICES DE CONFIDENTIALITE DES DONNEES

La confidentialité est que L'information contenue dans les objets ne doit être ni rendue accessible, ni divulguée, à un sujet non autorisé [NCS 87].
La solution venant en premier à l'esprit est de rendre incompréhensible les messages transitant entre l'émetteur et le destinataire : de cette manière, même si un intrus lit les messages, il ne peut en d'enduire des informations. Cette solution est facilement réalisable à partir du moment où les deux entités désirant communiquer partagent une clé secrète : il est alors possible d'utiliser un crypto système classique pour chiffrer les

messages. Cependant, il existe une autre approche qui consiste à utiliser un **canal sécurisé** et à envoyer les données sur ce canal. Dans ce cas on fait confiance à une infrastructure de transport et à la sécurisation de celle-ci (protocoles sûr) pour acheminer le message. C'est le cas des VPN (voire chapitre III) ou des sessions SSL (voire chapitre II).

Le chiffrement consiste à transformer des informations en clair ("clear text") en un texte chiffré ("cipher text") à l'aide d'une clé maintenue secrète et d'une fonction (réversible) de chiffrement. La fonction inverse s'appelle le déchiffrement.

Il existe deux classes d'algorithme de chiffrement : chiffrement symétrique et chiffrement asymétrique.

5.1.1 CHIFFREMENT SYMETRIQUE

Les algorithmes de chiffrement symétrique, comme par exemple DES [NBS 77] ou IDEA [LAI 92] utilisent la même clé pour le chiffrement et le déchiffrement.

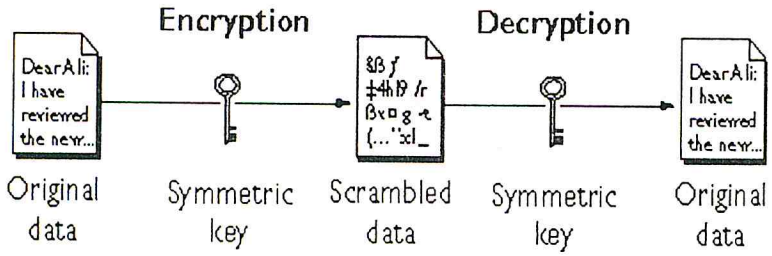


Figure 1. 5: chiffrement symétrique.

5.1.2 CHIFFREMENT ASYMETRIQUE

Le chiffrement asymétrique [DIF 76], utilise une clé pour le chiffrement et une autre pour le déchiffrement. La clé de chiffrement, qui est normalement connue par tout le monde (elle s'appelle aussi *la clé publique*), permet d'envoyer un message chiffré à quelqu'un qui connaît la clé de déchiffrement. Cette clé ne doit pas être divulguée par le récepteur, et elle s'appelle donc aussi *la clé secrète* [RIV 78].

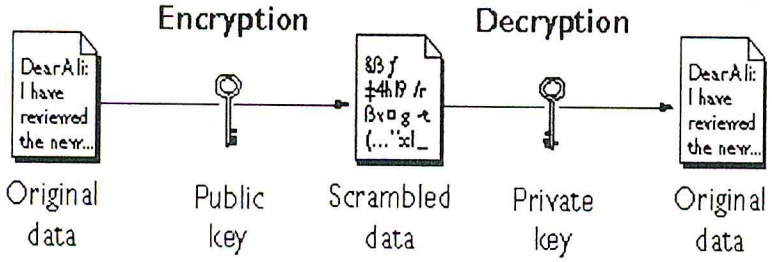


Figure 1. 6: chiffrement asymétrique.

5.1.3 PROTOCOLES SURS

Un protocole sûr est un protocole spécialement conçu pour protéger l'intégrité et la confidentialité de tous les objets qui sont transportés.

1. un mécanisme d'authentification est nécessaire pour s'assurer que la communication est toujours établie entre les entités correctes,
2. un protocole sécurisé doit être protégé contre la modification des paquets, la perte des paquets et la réutilisation d'anciens paquets. Par conséquent, des numéros de séquence et des « time Stamp » qui ne peuvent pas être altéré doivent être contenus dans tous les messages,
3. le protocole doit être protégé contre les attaques par déni de service ou la perte de paquets. Il est nécessaire que les parties communicantes échange des messages à des intervalles réguliers pour s'assurer que tous les messages ont été reçus.

Il existe un protocole «*IPsec*» qui est conforme à toutes les exigences ci-dessus. Il s'agit de la norme de sécurité de l'IPv6, cette norme met en oeuvre des services de sécurité (authentification bilatérale, confidentialité et intégrité) au niveau des paquets IP (on parle de sécurité couches basses). Elle présente donc l'avantage de permettre une sécurisation de toute communication passant sur IP, et de ce fait indépendante des protocoles applicatifs mis en oeuvre (HTTP, FTP...). IPSec est nativement inclus dans l'IPv6, cependant il peut être implémenté dans l'IPv4 utilisé aujourd'hui. C'est notamment le cas pour des applications comme les VPN (Virtual Private Network). Par ailleurs, IPSec peut aussi être utilisé au niveau applicatif, comme le fait la société SSH avec son logiciel éponyme [HPD 01].

5.2 SERVICES D'AUTHENTIFICATION

L'authentification doit assurer que l'accès au système doit être autorisé uniquement aux sujets habilités.

La phase d'authentification est décomposée en deux parties : l'identification, lors de laquelle l'utilisateur présente son identifiant, et l'authentification où l'utilisateur prouve son identité [BID 95]. Par exemple le système Unix maintient un fichier avec les mots de passe chiffrés de tous les utilisateurs enregistrés. La procédure de `login` exige que l'utilisateur spécifie son identité et son mot de passe, puis vérifie que les deux correspondent. Il y a deux problèmes avec ce schéma : les utilisateurs choisissent souvent des mots de passe faciles à deviner (comme "password" ou "secret") et les mots de passe sont vulnérables chaque fois qu'ils sont utilisés (beaucoup de systèmes transmettent les mots de passe en clair sur le réseau).

Pour répondre aux défaillances des mots de passe, un certain nombre de mécanismes d'authentification ont été proposés. La plupart des ces mécanismes se basent sur les protocole d'authentification suivants [NEE 78] :

5.2.1 RADIUS

RADIUS (Remote Authentication Dial-In User Service) ; Le protocole RADIUS fonctionne selon un modèle client/serveur.

- Le client RADIUS, il effectue des requêtes RADIUS et agit en fonction des réponses reçues,
- Le serveur RADIUS peut agir en tant que Proxy RADIUS pour les systèmes d'authentifications.

Les transactions RADIUS sont authentifiées par l'utilisation d'une clé secrète qui n'est jamais transmis sur le réseau. De plus les mots de passe sont encryptés en utilisant cette même clé secrète. Le protocole RADIUS utilise le protocole UDP sur le port 1812 [COZ 03].

5.2.2 TACACS

TACACS (Terminal Access Controller Access Control System) ; Le protocole TACACS est un protocole d'authentification, qui permet à un serveur d'accès à distance d'expédier le mot de passe dans la procédure de connexion d'un utilisateur à un serveur d'authentification pour déterminer si on peut permettre l'accès à un système donné. TACACS est un protocole non codé et donc moins sécurisé que le protocole RADIUS.

Une version postérieure de TACACS est TACACS+ (TACACS étendu). C'est un protocole entièrement nouveau. TACACS et RADIUS ont généralement remplacé les vieux protocoles. TACACS utilise TCP, pour cette raison quelques administrateurs recommandent d'utiliser TACACS puisque TCP est vu comme un protocole plus fiable. De plus, RADIUS combine l'authentification et l'autorisation dans un profil d'utilisateur, quant à lui TACACS sépare les deux [COZ 03].

On trouve aussi des logiciels d'authentification comme le Kerberos :

5.2.3 KERBEROS

Kerberos ou Cerbère est un système distribué d'authentification qui permet à un sujet à prouver son identité à un vérificateur (un serveur) sans envoyer des données à travers le réseau qui pourrait permettre à un agresseur de pirater ces données.

Kerberos fournit l'intégrité et la confidentialité pour des données envoyées entre le client et serveur.

Au départ, le client souhaite envoyer un message au serveur. Le serveur, pour être sûr de l'identité du client, lui demande de s'authentifier. Interviennent alors le serveur d'authentification et le vérificateur.

Le client et le serveur d'authentification possèdent la même clé d'authentification. Pour le client, cette clé est un mot de passe. Celui-ci envoie un message d'authentification crypté avec cette clé au serveur d'authentification. Celui-ci utilise alors sa clé pour vérifier l'identification du client. Si le message est décrypté, le client est authentifié.
Le service d'authentification envoie alors au client un certificat crypté contenant une clé unique permettant d'identifier le client (clé session), et une date d'expiration au delà de laquelle il ne pourra plus s'identifier, que par une clé connue seulement du vérificateur (clé serveur), alors le client ne peut donc pas modifier le certificat puisqu'il ne connaît pas cette clé.
Le certificat est alors acheminé jusqu'au vérificateur qui le décrypte en utilisant la clé serveur et utilise a clé session obtenue pour authentifier le client

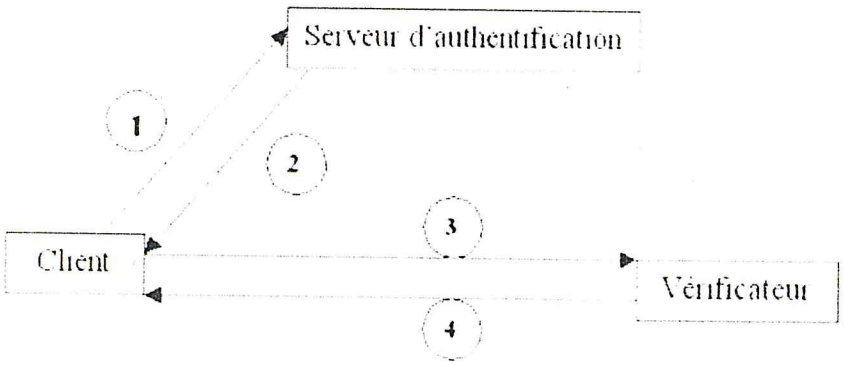


Figure 1. 7: **Fonctionnement de Kerberos** [COZ 03]

1. Envoi du message d'authentification crypté avec la clé client,
2. Si client authentifié, Le serveur d'authentification utilise sa clé serveur pour envoyer le certificat crypté contenant la clé session (clé unique identifiant le client),
3. Envoi de ce certificat crypté au vérificateur,
4. Le vérificateur, après avoir décrypté le certificat avec sa clé serveur, obtient la clé session qui permet d'authentifier le client [COZ 03].

5.3 SERVICES D'INTEGRITE

Les services d'intégrité des données aident à protéger les données et les logiciels et les autres composants de réseau, contre les modification non autorisées, lesquelles pouvant être de nature intentionnelle ou accidentelle. Ce type de service peut être assuré par l'utilisation de contrôle d'accès. Ces services contrecarrent les menaces actives en offrant une protection efficace contre la modification, l'insertion et l'effacement des données et des flux.
Un mécanisme de contrôle d'accès est implémenté dans la plupart des systèmes d'exploitation multi-utilisateurs (droit de lecture, écriture, suppression…).

Le contrôle d'accès peut être soit *discriminatoire,* soit *mandataire ;* nous précisons ces deux types d'accès dans les paragraphes suivants.

5.3.1 CONTROLE D'ACCES DISCRETIONNAIRE

La première technique de contrôle d'accès introduite fut la plus naturelle : un fichier appartient à son créateur (qui est alors considéré comme le propriétaire) qui, par conséquent, a le droit d'en faire ce qu'il veut. Ce contrôle d'accès est dit discrétionnaire [BID 95].

Le contrôle d'accès est dit discrétionnaire lorsque la technique de restriction d'accès aux objets est basée sur l'identité des sujets ou des groupes auxquels ils appartiennent. Le contrôle est discrétionnaire dans le sens où un sujet possédant un certain droit d'accès est capable de conférer ce droit à tout autre utilisateur [DOS 83].

Pour le cas d'un contrôle d'accès discrétionnaire, la gestion des informations sensibles (c'est à dire, les informations devant être soient confidentielles, soient intègres) est sous la responsabilité du propriétaire de ces informations. Par ailleurs, la politique de sécurité est individuelle, c'est à dire que chaque utilisateur construit sa propre politique de sécurité. Le modèle de sécurité consiste alors à vérifier que le système informatique applique correctement les droits d'accès spécifiés par chaque utilisateur [BID 95]. Remarquons que l'abus de cette confiance peut amener le système dans un état non sûr, (c'est à dire contraire à la politique d'autorisation définie) [JEN 99]. Ainsi les systèmes qui réalisent une politique d'autorisation discrétionnaire sont particulièrement vulnérables aux attaques comme le cheval de Troie.

5.3.2 CONTROLE D'ACCES MANDATAIRE

Dans le cas d'une politique d'autorisation d'accès mandataire, les interactions entre sujets et objets sont dirigées par des règles incontournables. Ces règles déterminent les droits d'accès qu'un sujet particulier peut posséder sur n'importe quel objet. Le contrôle d'accès mandataire est requis pour les systèmes à haut niveau de sécurité [DOS 83].

L'une des façons de spécifier ces règles est d'imposer une hiérarchie pour les sujets et pour les objets ; c'est le cas des politiques multi–niveaux appliquées par les militaires [LAN 81]. Dans une telle politique, les informations sont classées selon leur sensibilité et les utilisateurs sont habilités à accéder à l'information jusqu'à un certain niveau de classification de sécurité [JEN 99].

De nombreux schémas de contrôle d'accès mandataire sont présents dans la littérature : le premier est sans aucun doute le modèle de Bell-LaPadula [BEL 76], dont l'objectif est la confidentialité des données, et qui a été proposé pour résoudre les

problèmes de l'organisation de la défense américaine. Mais il existe aussi d'autres politiques mandataires, par exemple le modèle d'intégrité de Biba [BIB 77].
Dans les politiques mandataires, la confiance est remplacée par le contrôle. Comme dit le proverbe, «la confiance est bien mais le contrôle est mieux ».

5.4 SERVICES DE LA DISPONIBILITE

Comme leur nom l'indique, les services de disponibilité assurent à tous les utilisateurs l'accès aux ressources et aux données, comme prévu. Dans certaines applications, ces services visent expressément à conter les attaques ou les événements pouvant mener à un déni de service sur le réseau. Dans les organisations ou la perturbation des fonctions du réseau peut causer des graves préjudices, il ne sera probablement pas suffisant de faire régulièrement des copies de sécurité des données. Les services de disponibilité vont au-delà de la simple copie de sécurité. En fait, on peut diviser ces services en deux groupes principaux.
Dans le premier, nous retrouvons des services de contingentement, c'est à dire. Les services qui sont requis pour empêcher les personnes, que leurs intentions soient malicieuses ou non, de sur utiliser les ressources du réseau, comme l'espace disque, la mémoire, la largeur de bande, ... etc. de telle sorte que ces ressources ne soient plus disponibles pour les autres utilisateurs.
Par ailleurs, un deuxième groupe de services de disponibilité assure le maintien des fonctions du réseau lorsqu'il y a une panne de matériel ou de logiciel ; ce groupe est sous l'appellation « tolérance aux pannes ».

5.4.1 TOLERANCE AUX PANNES

Les services de tolérance aux pannes offrent au réseau la possibilité de résister aux défectuosités et aux pannes de composants, et de continuer à fonctionner pendant que l'on remplace les composants défectueux et (ou) que l'on récupère les données après une interruption de services. Ces services sont cruciaux, car ils maintiennent la fonctionnalité du réseau; voici quelques services de ce type :

- L'alimentation sans coupure,
- Les disques durs synchronisés et jumelés,
- La copie de sécurité des données,
- Archivage et copie de sécurité des clés de chiffrement.

6. CONCLUSION

La sécurité joue un rôle très particulier, parce que la moindre défaillance peut compromettre le bon fonctionnement du système. Si l'algorithme d'ordonnancement

marche dans 95% des cas, mais n'est pas équitable dans 5% des cas, le système continu à fonctionner. Si le système de sécurité ne marche que dans 95% des cas, les 5% restants peuvent être exploités par un adversaire et compromettre toute la sécurité du système.

Dans ce chapitre, nous avons identifié un certain nombre d'objectifs pour la sécurité et des menaces contre la sécurité des réseaux informatiques. Les objectifs les plus importants sont :

- La confidentialité des informations stockées dans le système,
- L'intégrité de ces informations,
- La disponibilité des informations et des ressources du système.

Pour atteindre ces objectifs, le système doit mettre un certain nombre de dispositifs en place, notamment les mécanismes pour :

- L'identification et l'authentification des sujets du système,
- Le contrôle d'accès aux ressources,
- La communication sûre.

Dans la suite de notre étude on va baser sur la sécurité de l'information, puisque elle concerne les systèmes informatiques. Le chapitre suivant présente avec détail les systèmes interconnectés et leurs vulnérabilités.

CHAPITRE II

INTERCONNEXIONS DES SYSTEM ET LEURS FAILLES

1. INTRODUCTION

L'évolution accélérée des systèmes informatiques et leurs intégrations dans tous les domaines est devenue une réalité quotidienne. L'intérêt de mettre en commun des fonctions de traitement a fait naître le besoin des réseaux.

La raison d'être d'un réseau est en effet le partage de ressources et la mise en disposition de matériels ou de services à plusieurs utilisateurs géographiquement repartis. Un réseau permet à des équipements de communiquer, et par la même, facilite la décentralisation des traitements.

On va voir dans ce chapitre les deux architectures des réseaux, et leurs failles.

2. LE MODELE DE REFERENCE OSI

l'interconnexion de systèmes ouverts (OSI : *Open Systems Interconnection*) est un ensemble de normes internationales ayant pour objet de donner à des applications informatiques, résidant dans des réseaux d'ordinateurs d'architectures internes différentes, les moyens de communiquer entre elles. Cette activité de normalisation a été lancée à la fin des années 1970 par l'ISO (*Organisation internationale de normalisation*) et le CCITT (*Comité Consultatif International Téléphonique et Télégraphique*) devenu depuis 1993 UIT-T (*Union Internationale des Télécommunications Normes de Télécommunication*). Les résultats atteints sont d'une importance considérable, tant par la conception d'une architecture de réseau normalisé que par la collection des normes ISO et de recommandations UIT-T [TAN 01].

Un système ouvert est un ordinateur, un terminal, un réseau ou n'importe quel équipement respectant cette norme est donc apte à échanger des informations avec d'autres équipements hétérogènes et issus de constructeurs différents. Le premier objectif de la norme OSI a été de définir un modèle de toute architecture de réseau, basé sur un découpage en **sept couches**, chacune de ces couches correspondant à une fonctionnalité particulière d'un réseau. Les couches 1, 2, 3 et 4 sont dites **basses** et les couches 5, 6 et 7 sont dites **hautes**. Chaque couche est constituée d'éléments matériels et logiciels et offre un service à la couche située immédiatement au-dessus d'elle en lui épargnant les détails d'implémentation nécessaires. Chaque couche (N) d'une machine gère la communication avec la couche (N) d'une autre machine en suivant un **protocole** de niveau (N) qui est un ensemble de règles de communication pour le service de niveau (N).

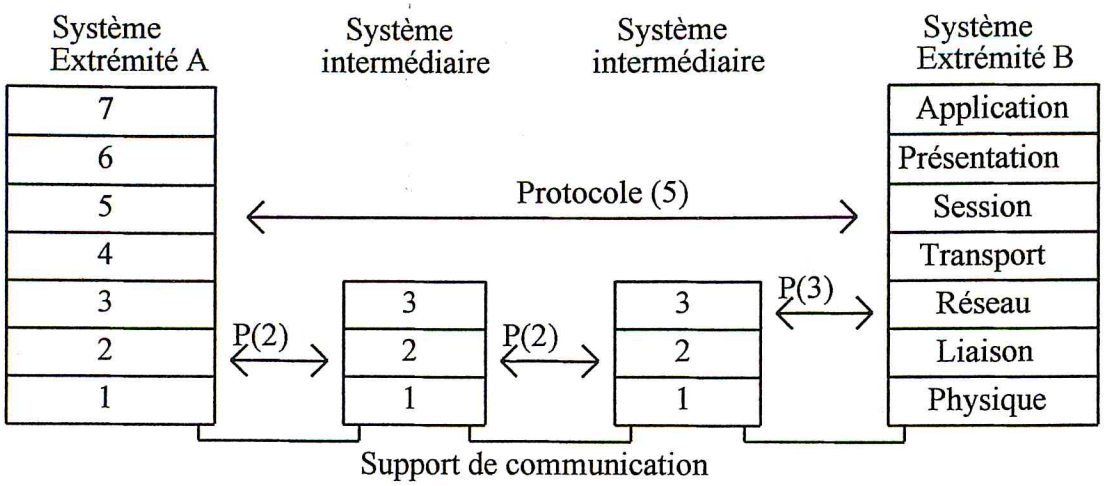


Figure 2. 1: le modèle OSI

En fait, aucune donnée n'est transférée directement d'une couche (N) vers une autre couche (N), mais elle l'est par étapes successives. Supposons un message à transmettre de l'émetteur A vers le récepteur B. Ce message, généré par une application de la machine A va franchir les couches successives de A via les interfaces qui existent entre chaque couche pour finalement atteindre le support physique. Là, il va transiter via différents nœud du réseau, chacun de ces nœuds traitant le message via ses couches basses. Puis, quand il arrive à destination, le message remonte les couches du récepteur B via les différentes interfaces et atteint l'application chargée de traiter le message reçu [PUJ 98].

2.1 MODES DE FONCTIONNEMENT

L'OSI reconnaît deux modes de fonctionnement avec et sans connexion.

En mode **avec connexion**, les entités entre lesquelles doit fonctionner un protocole de couche établissent explicitement une relation qui les lie jusqu'à ce qu'elles décident de rompre cette liaison. L'opération se déroule en trois phases :

- phase d'établissement de la connexion, au cours de laquelle les entités peuvent négocier entre elles les caractéristiques du protocole en choisissant, par exemple, des options particulières ;
- phase de transfert de données ;
- phases de libération de la connexion.

Une connexion établie dans une couche sert à mettre en relation entre elles des entités de la couche immédiatement supérieur.

En mode **sans connexion**, seule existe la phase de transfert de données. La mise en œuvre de ce mode est donc plus rapide, mais supposant de la part de chacun des partenaires une connaissance a priori des capacités de l'autre, puisqu'il n'y a pas de négociation possible. Par ailleurs, chaque transfert exige la répétition d'informations qui, dans le cas du mode avec connexion, peuvent être mise en communs dés le début.

Le choix du mode à utiliser est donc lié étroitement aux caractéristiques de la communication désirée [PUJ 98].

2.2 LA COUCHE PHYSIQUE

La couche physique s'occupe de la transmission des bits de façon brute sur un canal de communication. On doit connaître le nombre de volts à atteindre pour représenter un bit à 1 et à 0, la durée de transmission d'un bit en microsecondes, la possibilité de transmission dans les deux directions simultanément, l'initialisation de la connexion et le relâchement quand les deux cotés ont fini. Le nombre de broches, que possède le connecteur de réseau et le rôle de chacune d'entre elles. Les problèmes de conception concernent les interfaces mécanique, électrique et fonctionnelles ainsi que le support physique de transmission qui se trouve sous la couche physique [TAN 01].

2.2.1 LES MODES DE TRANSMISSION

- **Transmission en bande de base** : envoyer directement les suite de bits sur le support à l'aide de signaux carrés ;
- **Transmission modulée** : émet un signal sinusoïdal qui, même s'il est affaibli, sera facilement décodable par le récepteur. Ce signal sinusoïdal est obtenu grâce à un modem (*modulateur-démodulateur*).

2.2.2 Les supports de transmission

L'objectif de la couche 1 du modèle OSI est aussi de fixer les caractéristiques des matériels utilisés pour relier physiquement les équipements d'un réseau. Nous décrivons brièvement quelques uns des supports de transmission les plus usités[PUJ 98].

La **paire torsadée** (Câble RJ45) est un câble téléphonique constitué à l'origine de deux fils de cuivre isolés et enroulés l'un sur l'autre ;

Le **câble coaxial** (Câble BNC) est un câble utilisé également en téléphonie et en télévision, il est constitué d'un cœur qui est un fil de cuivre ;

La **fibre optique** est un support d'apparition plus récente mais son utilisation prend de l'ampleur de jour en jour car elle permet des débits de plusieurs Gbit/s sur de très longues distances [NIC 99].

2.2.3 ETHERNET

La plupart des réseaux locaux actuels utilisent Ethernet. Cette couche correspond aux couches 1 et 2 du modèle OSI. Nous allons donc décrire l'entête Ethernet. Lorsqu'une donnée est envoyée sur Ethernet, toutes les machines sur le réseau voient le paquet. Il faut donc une information pour que seule la machine concernée par le message l'enregistre. Chaque paquet Ethernet contient un entête de 14 octets qui contient les adresses source et destination. Chaque machine est supposée reconnaître les paquets contenant son adresse Ethernet dans le champ destinataire. Il y a donc des moyens de tricher et de lire les données du voisin, c'est une des raisons pour lesquelles Ethernet manque de sécurité. Notons qu'il n y'a pas de relation entre

l'adresse Ethernet et l'adresse Internet. Chaque machine doit posséder une table de correspondance entre adresse Ethernet et Internet. Le checksum est calculé sur le paquet entier et ne fait pas partie de l'entête. Il est mis à la fin du paquet. Si on note **E** l'entête Ethernet et **C** le checksum, notre fichier initial ressemblera à :

E (donnée...) **C E** (donnée...) **C E** (donnée...) C E (donnée...) **C...C** [ROU 94].

2.3 LA COUCHE LIASON DES DONNEES

La couche liaison de données doit réaliser un certain nombre de fonctions spécifiques. Elle offre une interface de service clairement définie à la couche réseau. Elle détermine la façon dont les bits venant de la couche physique sont regroupés en trames et se charge de traiter les erreurs de transmission. Elle effectue enfin un contrôle de flux pour régulariser le volume des données échangées entre entités source et destination.

Si la ligne est utilisée pour une transmission bidirectionnelle, le travail de gestion de la couche est encore plus compliqué. Le problème est que les trames d'acquittement de B vers A et les trames de données du trafic de A vers B se disputent l'usage de la liaison. Une solution astucieuse (la *superposition* ou *piggybacking*) a été inventée.

2.3.1 LA NOTION DE TRAMES

Pour que la couche liaison de donnée puisse détecter les erreurs, elle découpe le train de bits en trames et calcule le total de contrôle, pour cela on a plusieurs méthodes de découpage [PUJ 98]:

1. Compter les caractères ;
2. Avoir des caractères de début et de trame, de fin de trame et de transparence ;
3. Utiliser de fanions de début et de fin de trame avec des bits de transparence ;
4. Violer le codage normalement utilisé dans la couche physique.

2.3.2 LE CONTROLE DE FLUX

Le contrôle de flux consiste à gérer les trames pour qu'ils transitent le plus rapidement possible entre l'émetteur et le récepteur. Il cherche à éviter les problèmes de congestion du réseau qui surviennent lorsque trop de messages y circulent. On peut citer les méthodes suivantes :

- Dans le **contrôle par crédits**, seuls *N* paquets sont autorisés à circuler simultanément sur le réseau,
- **Mécanisme de fenêtre**. Les paquets de données sont numérotés modulo 8 et les stations contiennent deux compteurs : *P(S)* un compteur de paquets **émis** et *P(R)* un compteur de paquets **reçus**. L'émetteur n'est autorisé à émettre que les paquets inclus dans la fenêtre.

2.3.3 DETECTION ET CORRECTION D'ERREURS

Les techniques employées ici reposent sur l'utilisation de **codes correcteurs** ou **codes détecteurs** d'erreurs qui chacun transforme la suite de bits à envoyer en lui ajoutant de l'information à base de **bits de redondance** ou **bits de contrôle**. Le récepteur se sert de cette information ajoutée pour déterminer si une erreur s'est produite et pour la corriger si la technique employée le permet.

2.3.4 LE PROTOCOLE HDLC

HDLC (High level Data Link control) est un protocole orienté bit définit un ensemble de procédures normalisées par l'ISO pour des communications, aussi bien point à point que multipoint, half ou full-duplex, mais toujours entre une machine primaire et une (ou plusieurs) machine(s) secondaires. Les différents modes sont les suivants :

- le **mode ABM** (*Asynchronous Balanced Mode*) est un mode de réponse asynchrone équilibré utilisé sur une liaison full-duplex entre 2 machines uniquement (liaison point à point) qui ont chacune le statut de primaire et de secondaire. Le secondaire peut émettre sans avoir reçu de permission du primaire ;
- le **mode NRM** (*Normal Response Mode*) est utilisé sur une liaison half duplex et ici le secondaire ne peut transmettre que sur invitation du primaire ;
- le **mode ARM** (*Asynchronous Response Mode)*, connu également sous le nom LAP est utilisé sur une liaison half-duplex également, mais le secondaire peut émettre sans que le primaire l'ait sollicité. Ceci peut alors provoquer des problèmes, si primaire et secondaire veulent simultanément émettre des données.

Les trames échangées ont l'allure suivante [TAN 01]:

Il y 3 types de trames (a : données), (b : supervision) et (c : initialisation)

	1	3		1	3*(bits)*
(a)	0	Seq		P/F	Suivant
(b)	1	0	Type	P/F	Suivant
(c)	1	1	Type	P/F	Suivant

Figure 2. 2: Les trois types de trames HDLC [TAN 01]

Le **fanion** est égal à `01111110` et pour que la transparence au code soit possible, c'est à dire pour que la présence d'une suite de 6 bits à «1» dans les données ne soit pas interprétée comme un fanion, l'émetteur insère un «0» après chaque suite de 5 «1». Le récepteur supprime ce «0» supplémentaire après 5 «1» consécutifs de manière à restaurer le caractère réellement émis. Il existe trois types de trame distingués par les 2 premiers bits du champ de **commande** :

- Les trames (a) **d'information** contiennent des données en provenance, ou à destination, des couches supérieures ;

- Les trames (b) de **supervision** assurent le contrôle d'erreur et de flux ;
- Les trames (c) **non numérotées** servent à l'initialisation de la liaison et aux problèmes de reprise sur erreur non réglés à la couche 2.

Le contrôle est assuré par la technique du **polynôme générateur** de la norme *V41*. Le protocole utilise une **fenêtre à anticipation**, avec un numéro de séquence codé sur 3 bits. 7 trames peuvent être en instance d'acquittement à tout moment. Le champ **Seq** est le numéro de séquence de la trame.

Le **bit P/F** signifie *Poll/Final* (Invitation à émettre/Fin). Il est utilisé quand un ordinateur interroge un groupe de terminaux. La valeur **P** indique que l'ordinateur invite un terminal à envoyer ses données. Toutes les trames envoyées par le terminal, sauf la dernière, ont le bit P/F mis à P. ce bit est mis à **F** dans la dernière trame.

Le champ Suivant est utilisé pour acquitter les trames reçues. Tous les protocoles ont adopté la convention selon laquelle la valeur de ce champ n'est pas le numéro de la dernière trame reçue correctement, mais celui de la première non encore reçue. Ce choix n'est pas arbitraire et ne porte pas à conséquence, mais il faut qu'il soit respecté par tous [TAN01].

2.3.5 LA SOUS-COUCHE DE CONTROLE D'ACCES AU CANAL

Les réseaux peuvent être divisés en deux catégories selon qui utilisent des connexions de type point à point où qu'ils effectuent des diffusions sur un réseau multipoint.

Dans les réseaux à diffusion, tous les abonnés ont la possibilité d'émettre et de recevoir. Le problème majeur consiste à déterminer qui a le droit d'émettre. Les protocoles utilisés pour déterminer qui sera le prochain élu d'un canal de communication à accès multiples sont regroupés dans une sous-couche interne à la couche liaison de données appelée sous-couche de contrôle d'accès au canal ou sous-couche **MAC** (*Medium Access Control*). Cette sous-couche joue un rôle très important dans les réseaux LAN, et plus particulièrement dans ceux dont le fonctionnement repose sur le principe de l'accès multiple. En revanche, dans les réseaux WAN on utilise généralement des liaisons point-à-point [TAN 01].

2.3.6 CSMA/CD

Le protocole CSMA/CD (Carrier Sense Multiples Access), ainsi que plusieurs autres protocoles de réseaux LAN, utilisent le modèle conceptuel représenté à la figure suivante. À l'instant marqué t0, une station termine la transmission de sa trame. Toutes les stations qui ont une trame à transmettre peuvent dés lors tenter de le faire. Si deux ou plusieurs stations commencent à transmettre simultanément, il se produit une collision, les collisions sont détectées en examinant le niveau électrique ou la largeur

des impulsions des signaux reçus (lors de l'écoute) et en comparant à ceux des signaux transmis [TAN 01].

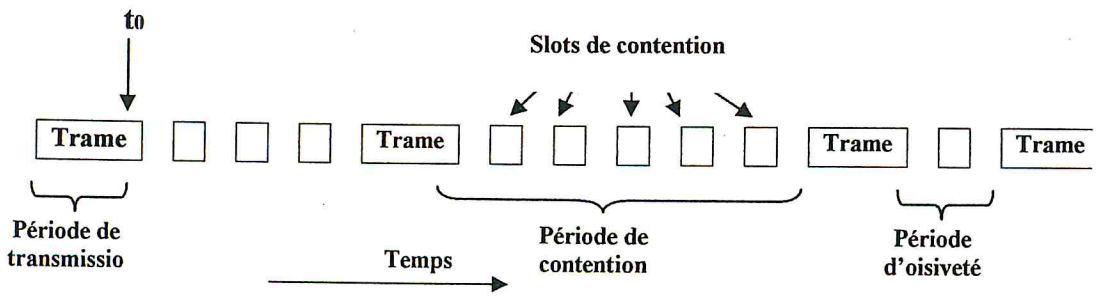


Figure 2. 3: Succession de différents états du protocole CSMA/CD [TAN 01]

Dés qu'une station détecte une collision, elle arrête sa transmission et laisse s'écouler un temps de durée aléatoire avant de tenter une nouvelle transmission, en s'assurant auparavant qu'aucun autre station ne l'a précédée dans la même démarche. Il résulte que le modèle CSMA/CD comporte une succession de périodes de transmissions et de contention entrecoupées de périodes d'oisiveté, le canal étant alors disponible.
Il est important de prendre conscience que la détection des collisions est un phénomène analogique. Le récepteur de la station doit écouter en permanence ce qui se passe sur le câble pendant que son partenaire, l'émetteur, transmet les informations.
Si ce qu'il reçoit est différent de ce qu'il transmis, il ne début pas si les signaux sont perturbés par une collision. Il en résulte que l'encodage des signaux doit permettre la détection de toutes les collisions : par exemple, la collision de deux signaux à 0 volt est indétectable et c'et pour cette raison qu'un encodage spécifique est couramment utilisé (encodage *Manchester*).

2.4 LA COUCHE RESEAU :

La couche réseau permet de gérer le sous-réseau. La façon dont les paquets sont acheminés de la source au destinataire construite un élément clé de sa conception. Les routes pouvant être fondées sur des tables **statiques** (définis par l'administrateur) dans le réseau est rarement changées. Elles peuvent également être **dynamiques**, déterminées au début de chaque conversation, recalculées pour chaque paquet de manière à prendre en compte la charge instantanée du réseau utiliser.
Si trop de paquets se trouvent simultanément dans le sous-réseau, il va se créer des engorgements. Le contrôle d'une telle **congestion** est aussi un domaine de la couche réseau [TAN 01].

2.4.1 ARCHITECTURE DE LA COUCHE RESEAU

Les détails complémentaire sur l'architecture de cette couche sont spécifies dans la norme ISO/CEI 8648, architecture interne de la couche réseau. La complexité de cette couche provient de la nécessité de concevoir une représentation générale applicable à toutes les sortes de supports de transmission envisageables.

La solution passe par la notion de **sous-réseau**, c'est une représentation de tout ensemble homogène de moyens de transmission. Un sous-réseau possède des mécanismes internes d'adressage, d'acheminement de l'information entre eux.

Il n'est pas nécessaire que les sous réseaux soient totalement connectés entre eux, il suffit de définir au moins un chemin entre toute paire de système d'extrémité.

Architecturalement, les fonctions additionnelles peuvent être représentées sous la forme d'une structure indépendante, l'unité **d'interfonctionnement**, située entre deux ou plusieurs sous réseaux On y trouve, dans la mesure des besoins particuliers :

- des protocoles **d'accès** à chacun des sous réseaux auxquels elle est reliée ;
- des protocoles **complémentaires**, appelés protocoles de **convergence dépendants** du sous-réseau, chargés d'apporter les fonctions manquantes à la réalisation du service du réseau OSI ;
- des protocoles de **convergence indépendants** du sous-réseau qui véhiculent, en particulier, les informations nécessaires à la mise à jour des tables d'acheminement.

Il y a fondamentalement deux philosophies différentes dans l'organisation des sous réseaux, l'une utilisant le mode avec connexion et l'autre travaillant dans le mode sans connexion.

Dans le contexte des opérations internes du sous-réseau une connexion appelée **circuit virtuel**, par analogie avec les circuits physiques du système téléphonique.

Les paquets indépendants du monde non connecté sont appelés **datagrammes** par analogie avec le télégramme [TAN 01].

Tableau 1 : Comparaison des datagramme et circuit virtuel [TAN 01].

Caractéristiques	Sous-réseau datagramme	Sous-réseau circuit virtuel
Etablissement du circuit	Pas nécessaire	nécessaire
Adressage	Chaque paquet contient les adresses complètes de la source et du destinataire.	Chaque paquet contient le numéro de circuit virtuel.
Information de routage	Le sous-réseau ne conserve aucune information de routage des paquets.	Chaque circuit virtuel établi requiert de la place dans les tables de routages.
Routage	Chaque paquet a un routage indépendant.	La route est établie à l'initialisation du CV ; chaque paquet suit cette route.
Conséquences d'une défaillance de routeur.	Aucune ; sauf la perte des paquets présents dans le routeur incriminé.	Tous les circuits virtuels traversant l'équipement défaillant sont détruits.
Contrôle de congestion	Difficile et complexe.	Facile lorsqu'il est possible d'allouer suffisamment d'espace mémoire pour établissement du circuit virtuel.

physique et la couche liaison de données de l'OSI sont combinées en une couche réseau [FES 99].

L'architecture de TCP/IP comporte 4 couches (application, transport ou hôte à hôte, Internet et la couche accès réseau).

Comme on peut le remarquer, les couches du modèle TCP/IP ont des tâches beaucoup plus diverses que les couches du modèle OSI, étant donné que certaines couches du modèle TCP/IP correspondent à plusieurs couches du modèle OSI.

Les rôles des différentes couches sont les suivants [NIC 99]:

- **Couche accès réseau** spécifie la forme sous laquelle les données doivent être acheminées quelque soit le type de réseau utiliser (Ethernet, anneau à jeton, FTS, FDDI, PPP ...)
- **Couche Internet** : elle est chargée de fournir le paquet de données (datagramme) avec les protocoles IP, ARP, RARP ...
- **Couche Transport** : elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaître l'état de la transmission en utilisent le protocole TCP ou UDP
- **Couche Application** : elle englobe les applications standard du réseau (Telnet, SMTP, FTP, ...)

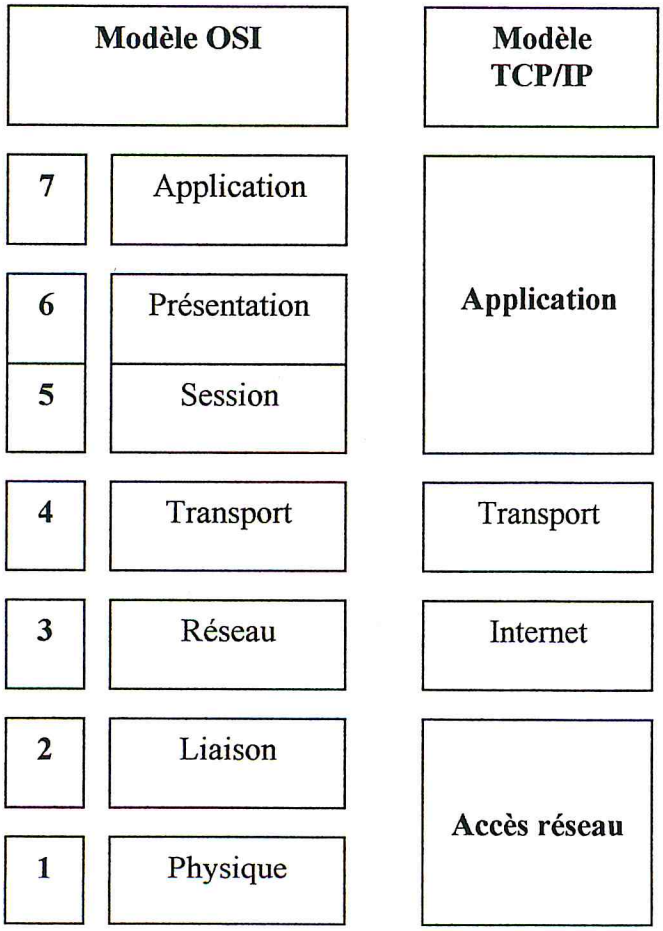


Figure 2. 4: OSI et TCP/IP [FES 99]

Lors d'une transmission, les données traversent chacune des couches au niveau de la machine émettrice. A chaque couche, une information est ajoutée au paquet de données, il s'agit d'un **en-tête**, ensemble d'informations qui garantissent la transmission. Au niveau de la machine réceptrice, lors du passage dans chaque couche, l'en-tête est lu, puis supprimé. Ainsi, à la réception, le message est dans son état originel...

Les termes utilisés pour référencer les données au niveau de chaque couche de TCP/IP sont illustrés ci dessous :

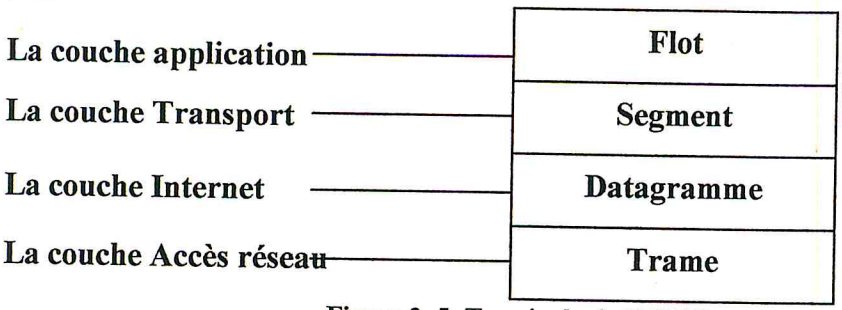


Figure 2. 5: Terminologie TCP/IP

3.1 LA COUCHE ACCES RESEAU

La couche accès réseau est la première couche de la pile TCP/IP, elle offre les capacités à accéder à un réseau physique quel qu'il soit, c'est à dire les moyens à mettre en oeuvre afin de transmettre des données via un réseau. Ainsi, cette couche contient toutes les spécifications concernant la transmission de données sur un réseau physique, qu'il s'agisse de réseau local (Anneau à jeton, Ethernet, FDDI), de connexion à une ligne téléphonique ou n'importe quelle type de liaison à un réseau. Elle prend en charge les notions suivantes [PIL 03] :

- Acheminement des données sur la liaison
- Coordination de la transmission de données (synchronisation)
- Format des données
- Conversion des signaux (analogique/numérique)
- Contrôle des erreurs à l'arrivée

Toutes ces spécifications sont transparentes aux yeux de l'utilisateur, car l'ensemble de ces tâches est en fait réalisé par le système d'exploitation, ainsi que les drivers du matériel permettant la connexion au réseau (ex : driver de carte réseau).

3.1.1 La liaison point À point

Par la ligne téléphonique classique, deux ordinateurs maximum peuvent communiquer par modem ensemble, au même titre qu'il n'est pas possible d'appeler simultanément deux personnes par la même ligne téléphonique. On dit alors que l'on a une **liaison point à point**, c'est à dire une liaison entre deux machines réduite à sa plus

simple expression: il n'y a pas nécessité de partager la ligne entre plusieurs machines, chacune parle et répond à son tour.

Ainsi, de nombreux protocoles de modem ont été mis au point. Les premiers d'entre eux permettaient une simple transmission de données entre deux machines, puis certains furent dotés d'un contrôle d'erreur, et avec la montée d'Internet, ils furent dotés de la capacité d'adresser des machines. De cette façon, il existe désormais deux grands protocoles de modem :

- SLIP : un protocole ancien, faible en contrôles
- PPP : le protocole le plus utilisé pour les accès à Internet par modem, il autorise un adressage des machines

3.1.2 Protocole SLIP

SLIP (*Serial Link Internet* Protocol), protocole Internet de liaison en série. SLIP est le résultat de l'intégration des protocoles modems précédent à la suite de protocoles TCP/IP. Il s'agit d'un protocole de liaison Internet simple n'effectuant ni contrôle d'adresse, ni contrôle s'erreur, c'est la raison pour laquelle il est vite devenu désuet par rapport à PPP.

La transmission de données avec SLIP est très simple : ce protocole envoie une trame composée uniquement des données à envoyer suivies d'un caractère de fin de transmission (le caractère *END*, dont le code ASCII est 192). [PIL 03]

Une trame SLIP ressemble donc à ceci : **Données à transmettre | END**

3.1.3 Protocole PPP

PPP (Point to Point Protocol), protocole point à point. Il s'agit d'un protocole beaucoup plus élaboré que SLIP, dans la mesure où il transfère des données supplémentaires, mieux adaptées à la transmission de données sur Internet (l'ajout d'informations dans une trame est en grande partie dû à l'augmentation de la bande passante).

PPP est en réalité un ensemble de trois protocoles :

- un protocole d'encapsulation de datagrammes
- un protocole de contrôle de liaison (**LCP**, Link Control Protocol), permettant des contrôles de test et de configuration de la communication
- un ensemble de protocoles de contrôle de réseau (**NCP**, Network Control Protocol), permettant des contrôles d'intégration de PPP au sein de protocoles de couches supérieures.

Les données encapsulées dans une trame PPP sont appelées *paquets*. Ces paquets sont généralement des datagrammes, mais il peut s'avérer qu'ils soient autres (d'où la dénomination spécifique de *paquet* au lieu de datagramme). Ainsi, un champ de la trame est réservé au type de protocole auquel le paquet appartient. Une trame PPP ressemble à ceci: **Protocole (1-2octets) | Données à transmettre | Données de remplissage**

Les données de remplissage servent à adapter la longueur de la trame pour certains protocoles [PIL 03].

Une session PPP se déroule comme suit:

- Lors de la connexion, un paquet LCP est envoyé en cas de demande d'authentification de la part du serveur, un paquet correspondant à un protocole d'authentification peut être envoyé (PAP,*Password Authentification Protocol*, ou CHAP,*Challenge Handshake Authentification Protocol* ou Kerberos) ;
- Une fois la communication établie, PPP envoie des informations de configuration grâce au protocole NCP ;
- Les datagrammes à envoyer sont transmis sous forme de paquets ;
- A la déconnexion, un paquet LCP est envoyé pour clôturer la session.

3.1.4 Protocole PAP

PAP (Password Authentication Protocol) est un protocole bidirectionnel simultané pour les paquets de transfert entre les parties dans un réseau. PAP comprend l'ordonnançant de données, le contrôle d'écoulement, la responsabilité, et la détection et reprise d'erreur. PAP est un procédé employé par des serveurs de PPP pour valider une demande de connexion. Le PAP fonctionne comme suit:

- Après que le lien soit établi, le demandeur envoie un mot de passe et une identification au serveur.
- Le serveur valide la demande et renvoie un accusé de réception, termine la connexion, ou offre au demandeur une autre chance. Des mots de passe sont envoyés sans sécurité et le créateur peut faire des tentatives répétées d'accès. Pour ces raisons, un serveur qui comprend CHAP offrira d'utiliser ce protocole avant d'utiliser PAP [GUI 00].

3.1.5 Protocole d'authentification CHAP

CHAP (Challenge-Handshake Authentication Protocol), est une procédure plus sécurisée pour se relier à un système que le procédé d'authentification de mot de passe PAP.

Fonctionnement **(Authentification en mode défi-réponse) :**

- Après que le lien soit fait, le serveur envoie un message au demandeur de connexion. Le demandeur répond avec une valeur obtenue en utilisant une fonction à sens unique d'informations parasites ;
- Le serveur contrôle la réponse en la comparant à son propre calcul de la valeur prévue d'informations parasites ;
- Si les valeurs s'assortissent, l'authentification est reconnue; autrement la connexion est terminée. À tout moment, le serveur peut inviter la partie reliée pour envoyer un nouveau message. Puisque des identificateurs de CHAP sont changés fréquemment et parce que l'authentification peut être demandée par le serveur à tout moment, CHAP fournit plus de sécurité que PAP [GUI 00].

3.2 LA COUCHE INTERNET

La couche située au-dessue de la couche Interface dans la hiérarchie des protocoles est la couche Internet. Le protocole Internet est au cœur de TCP/IP et le plus importent de la couche Internet. IP fournit les services de livraison des paquets de base sur lesquels les réseaux TCP/IP sont construits. La couche Internet assure les tâches suivantes :

- Un niveau de service commun indépendant des supports physique interconnectés ;
- Un mécanisme d'adresse global ;
- Un modèle de routage pour transférer les données à travers l'interconnexion de réseaux physique ;
- Un mécanisme de fragmentation et de reassemblage des paquets échangés (la taille maximale d'un datagramme est 65535 octets).

Toutes ces fonctionalités sont fondamentales pour mettre en œuvre le concept d'interconnexion de réseaux. Ceci permet aux équipement de réseaux de voir une interconnexion comme étant homogène de nature.

La couche Internet contient ces principaux protocoles [PIL 03]:

- Le protocole IP
- Le protocole ARP
- Le protocole ICMP
- Le protocole RARP
- Le protocole BGP, RIP, OSPF

3.2.1 Protocole IP

L'Internet Protocol (IP) est un protocole fondamental du modèle OSI et une partie intégrante de TCP/IP. IP est très adapté à tout réseau nécessitant un protocole efficace pour les communications entre machines.

Le protocole IP est un protocole :

- Non fiable, il n'assure pas la retransmission des datagrammes en cas d'erreurs ;
- Non connecté, il ne se soucie pas de savoir par quél nœud passe un datagramme, ni même des machines de départ et de destination ;
- Il n'assure pas le contrôle de flux.

Les tâches principales de ce protocole, consistent à :

- Adresse des datagrammes d'information d'un ordinateur à un autre ;
- Remise des datagrammes à la couche Interface ;
- Router les datagrammes en détermination où ils seront envoyés, et en proposant des itinéraires de substitution en cas de problèmes.

L'IP est un des protocoles les plus importants d'Internet car il permet l'élaboration et le transport des datagrammes IP (les paquets de données), sans toutefois en assurer la

"livraison". En réalité le protocole IP traite les datagrammes IP indépendamment les uns des autres en définissant leur représentation, leur routage et leur expédition.
Le protocole IP détermine le destinataire du message grâce à 3 champs :

- Le champ adresse IP : adresse de la machine;
- Le champ masque de sous-réseau : un masque de sous-réseau permet au protocole IP de déterminer la partie de l'adresse IP qui concerne le réseau;
- Le champ passerelle par défaut : Permet au protocole Internet de savoir à quelle machine remettre le datagramme si jamais la machine de destination n'est pas sur le réseau local.

Les données circulent sur Internet sous forme de datagrammes ; Voici ce à quoi ressemble un datagramme :

<table>
<tr><td colspan="4"><-- 32 bits --></td></tr>
<tr><td>Version</td><td>Longueur d'en-tête</td><td>type de service</td><td>Longueur totale</td></tr>
<tr><td colspan="3">Identification</td><td>Drapeau | Décalage fragment</td></tr>
<tr><td>Durée de vie</td><td colspan="2">Protocole</td><td>Somme de contrôle en-tête</td></tr>
<tr><td colspan="4">Adresse IP source</td></tr>
<tr><td colspan="4">Adresse IP destination</td></tr>
<tr><td colspan="2">Option</td><td colspan="2">Rembourrage</td></tr>
<tr><td colspan="4">Données</td></tr>
</table>

Figure 2. 6: datagramme IP [PIL 03]

Voici la signification des différents champs :

- **Version** : il s'agit de la version du protocole IP que l'on utilise (actuellement on utilise la version 4 *IPv4*) afin de vérifier la validité du datagramme. Elle est codée sur 4 bits;
- **Longueur d'en-tête** : il s'agit du nombre de mots de 32 bits sur lesquels est répartis l'en-tête;
- **Type de service** : il indique la façon selon laquelle le datagramme doit être traité;
- **Longueur totale** : il indique la taille totale du datagramme en octets. La taille de ce champ étant de 2 octets, la taille totale du datagramme ne peut dépasser 65536 octets. Utilisé conjointement avec la taille de l'en-tête, ce champ permet de déterminer où sont situées les données ;
- **identification, drapeaux (flags) et déplacement de fragment** sont des champs qui permettent la fragmentation des datagrammes, ils seront expliqués plus loin;
- **Durée de vie** : (appelée aussi TTL : *Time To Live*) indique le nombre maximal de routeurs à travers lesquels le datagramme peut passer. Ainsi ce champ est décrémenté à chaque passage dans un routeur, lorsque celui-ci atteint la valeur critique de 0, le routeur détruit le datagramme. Cela évite l'encombrement du réseau par les datagrammes perdus ;
- **Protocole** : ce champ permet de savoir de quel protocole est issu le datagramme

Par example on a ICMP: 1, IGMP: 2, TCP: 6, UDP: 17;

- **Somme de contrôle de l'en-tête (*header checksum*)** : ce champ contient une valeur codée sur 16 bits qui permet de contrôler l'intégrité de l'en-tête afin de déterminer si celui-ci n'a pas été modifié ; altéré pendant la transmission. La somme de contrôle est le complément à un de tous les mots de 16 bits de l'en-tête (champ *somme de contrôle exclu*). Celle-ci est en fait telle que lorsque l'on fait la somme des champs de l'en-tête (somme de contrôle inclue), on obtient un nombre avec tous les bits positionnés à 1 ;
- **Adresse IP Source** : Ce champ représente l'adresse IP de la machine émettrice, sur 32bits, il permet au destinataire de répondre ;
- **Adresse IP destination** : Adresse IP du destinataire du message sur 32bits ;
- **Option** : de taille variable si les datagrammes contiennent des options
- **Rembourrage** : Utilisé en cas d'option pour amener, si besoin, la longueur de l'entête Internet à un multiple de 32 bits [PIL 03].

a) Adresse IP

IP utilise une adresse sur 32 bits pour identifier la connexion de la machine, cette adresse contient une partie réseau et une partie hôte. Le nombre de bits d'adresse, utilisées pour identifier l'hôte, varie selon la classe des adresse. Il existe 3 classes principales et une pour le multicast

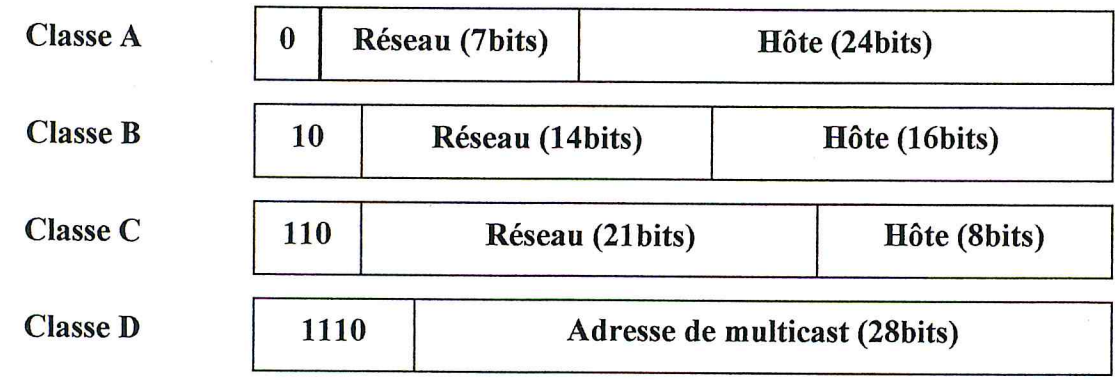


Figure 2. 7 : Les classes de IP [TIL 99]

Toutes les adresses hôtes et les adresses serveurs ne sont pas disponibles, dans toutes les classes, les adresses hôtes '0' et '255' sont réservées.
Si tous les bits de l'adresse hôte sont à '0', cette adresse identifie le réseau lui-même.
Si tous les bits sont à '1', cette adresse signifie un broadcast (diffusion), elle adresse simultanément tous les hôtes du réseau [RFC 1918].
Une entreprise qui décide d'utiliser des adresses à l'intérieur des plages spécifiées dans ce document peut le faire sans en référer à un organisme d'enregistrement. L'espace ainsi défini peut être simultanément utilisé par de nombreuses entreprises. Les adresses dans ces plages ne seront uniques qu'à l'intérieur de l'entreprise, ou du groupe

d'entreprises qui décident de se mettre d'accord sur cet espace d'adressage pour être capables de communiquer ensemble dans leur propre inter réseau.
Le service de remise du protocole IP est sans connexion. Les données utilisateurs sont envoyées sous forme datagramme [TIL 99].

b) La fragmentation des datagrammes IP

La taille d'un datagramme maximale est de 65535 octets. Toutefois cette valeur n'est jamais atteinte car les réseaux n'ont pas une capacité suffisante pour envoyer de si gros paquets. De plus, les réseaux sur Internet utilisent différentes technologies, si bien que la taille maximale d'un datagramme varie suivant le type de réseau. La taille maximale d'une trame est appelée *MTU* (Maximum Transfer Unit), elle entraînera la fragmentation du datagramme si celui-ci a une taille plus importante que le MTU du réseau

Type de réseau	MTU (en octets)
Arpanet	**1000**
Ethernet	**1500**
FDDI	**4470**

Tableau 2 : les différentes tailles des MTU [PIL 03]

La fragmentation d'un datagramme se fait au niveau des routeurs, c'est à dire lors de la transition d'un réseau dont le MTU est important à un réseau dont le MTU est plus faible. Si le datagramme est trop grand pour passer sur le réseau, le routeur va le fragmenter, c'est à dire le découper en fragments de tailles inférieures au MTU du réseau et de telle façon que la taille du fragment soit un multiple de 8 octets.
Le routeur va ensuite envoyer ces fragments de manière indépendante et réencapsulé (il ajoute un en-tête à chaque fragment) de telle façon à tenir compte de la nouvelle taille du fragment, et en ajoutant des informations afin que la machine de destination puisse réassembler les fragments dans le bon ordre (rien ne dit que les fragments vont arriver dans le bon ordre étant donné qu'ils sont acheminés indépendamment les uns des autres...).
Pour tenir compte de la fragmentation, chaque datagramme possède plusieurs champs permettant leur réassemblage :

- **champ déplacement de fragment** : champ permettant de connaître la position du début du fragment dans le datagramme initial ;
- **champ identification** : numéro attribué à chaque fragment afin de permettre leur réassemblage dans le bon ordre ;
- **champ longueur total** : il est recalculé pour chaque fragments ;
- **champ drapeau** : il est composé de trois bits :
  - Le premier n'est pas utilisé,
  - Le second (appelé **DF** : *Don't Fragment*) indique si le datagramme peut être fragmenté ou non. Si jamais un datagramme a ce bit positionné à un, le

routeur ne peut pas l'acheminer sans le fragmenter, alors le datagramme est rejeté avec un message d'erreur,

- Le dernier (appelé **MF** : *More Fragments, Fragments à suivre*) indique si le datagramme est un fragment de donnée. Si l'indicateur est à zéro, cela indique que le fragment est le dernier (donc que le routeur devrait être en possession de tous les fragments précédents) ou bien que le datagramme n'ait pas fait l'objet d'une fragmentation [PIL 03].

c) Attaque sur IP

Il existe de nombreux moyens d'utiliser la fragmentation pour infiltrer un réseau et y causer un déni de service.

- ***Ping O'Death :*** Cette attaque crée un déni de service en utilisant un système de ping pour créer un paquet IP, qui dépassera la taille maximum autorisé pour un paquet IP. Les données vont arriver sur la machine destinataire sous la forme de petits paquets qui respectent la norme. Ce n'est qu'une fois assembler que ces paquets dépassent la taille maximum des datagrammes IP, ce qui peut entraîner le crash de la machine ;
- ***Tiny fragments :*** Cette attaque consiste à fragmenter sur deux paquets une demande de connexion TCP. Le premier paquet ne contient que les huit premiers octets de l'entête TCP. Les données du second paquet IP renferment la demande de connexion TCP. Or, les filtres IP appliquent la même règle de filtrage à tous les fragments d'un même paquet. Le filtrage du premier fragment détermine cette règle, qui est ensuite appliquée aux autres fragments sans plus 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 malgré le filtre IP ;
- ***Fragment overlapping :*** Si deux fragments IP se superposent, le deuxième écrase le premier. L'attaque consiste à forger deux fragments d'un paquet IP. Le premier fragment est accepté par le filtre IP, car il ne contient pas de demande de connexion TCP. Comme la règle de filtrage des paquets s'applique à tous les fragments du même paquet, le deuxième fragment, qui contient les données, est accepté par le filtre. Lors de la défragmentation, les données du deuxième fragment écrasent celles du premier à partir de la fin du huitième octet. Le paquet réassemblé constitue alors une demande de connexion valide pour la machine cible, et, là encore, la connexion s'établit malgré le filtre IP [TIL 99] ;
- **L'attaque Oshare :** elle consiste à envoyer une entête IP invalide à la victime. cette entête IP est rendue invalide en jouant sur la valeur des champs de l'entête qui spécifient la longueur du datagramme. Ce sont les champs « IHL » (la longueur du l'entête) et « longueur total », qui sont modifiés pour cette attaque ;

- **L'IP spoofing:** usurpation d'adresse IP. L'adresse IP d'un ordinateur est l'adresse qui est utilisée pour reconnaître un ordinateur sur internet. Un des principaux problèmes est qu'en utilisant le routage source d'IP, l'ordinateur du hacker peut se faire passer pour un ordinateur connu. Le routage source d'IP est une option qui peut être utilisée pour spécifier une route directe à une destination et renvoyer le chemin de retour à l'expéditeur. La route peut inclure l'utilisation d'autres routeurs ou de serveurs qui n'auraient normalement pas été utilisés pour faire suivre les paquets à la destination finale.

3.2.2 Protocole ARP

Le passage des datagramme de la couche IP vers la couche interface réseaux nécessite une convertion d'adresses, car IP utilise des adresses sur 4 octets(adresse IP) tandis que la couche interface utilise un mécanisme à 6 octets (adresse physique). Pour cela, ARP apporte une solution efficace pour la résolution des adresses IP en adresses physiques. La tâche d'ARP est de convertir une adresse IP en une adresse physique locale ou de réseaux, pour éviter aux applications de devoir les connaître. ARP contient essentiellement une table contenant la liste des adresse IP et des adresses physique correspondantes, cette table est appelée le cache ARP. La figure 2.8 montre son agencement où chaque ligne correspond à un composant et contient 4 informations le concernant.

- Index IF : le numéro de l'interface physique ;
- Adresse physique : c'est l'adresse physique du composant ;
- Adresse IP : l'adresse IP correspondant à l'adresse physique ;
- Type : c'est le type d'entrée dans la couche ARP.

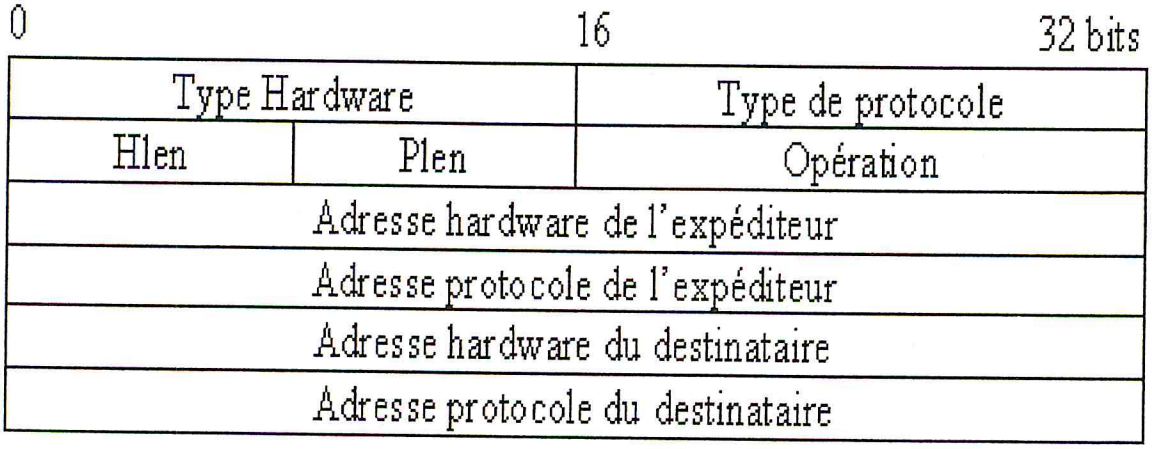


Figure 2. 8 : La structure d'une trame ARP [TIL 99]

- **Type Hardware** : spécifie le type de l'interface hardware ;
- **Type protocole** : spécifie le type du protocole de haut niveau émis par l'expéditeur;
- **Hlen** : longueur de l'adresse hardware ;
- **Plen** : longueur de l'adresse de haut niveau ;

- **Opération** : type de l'opération effectuée :
  1 Requête ARP,
  2 Réponse ARP,
  3 Requête RARP,
  4 Réponse RARP,
  5 Requête RARP dynamique,
  6 Réponse RARP dynamique,
  7 Erreur RARP dynamique,
  8 Requête InARP,
  9 Réponse InARP ;
- **Adresse hardware de l'expéditeur** : explicite ;
- **Adresse protocole de l'expéditeur** : explicite ;
- **Adresse hardware du destinataire** : explicite ;
- **Adresse protocole du destinataire** : explicite.

L'attaque ARP redirect

Cette attaque vise les réseaux locaux ethernet. C'est une technique de spoofing efficace mais détectable dans les logs utilisés pour la surveillance du réseau. Cette attaque consiste à s'attribuer l'adresse IP de la machine cible. On fait correspondre son adresse IP à l'adresse MAC de la machine pirate dans les tables de correspondances ARP des machines du réseau. Pour cela, il suffit d'envoyer régulièrement des paquets ARP_reply en broadcast, contenant l'adresse IP cible et la fausse adresse MAC. L'effet est de modifier les tables dynamiques de toutes les machines du réseau. Lorsqu'elles voudront communiquer avec la machine cible, les machines du réseau enverront leurs trames Ethernet à la machine pirate. A noter que les switches du réseau ne se rendent compte de rien. De son côté, la machine pirate stocke le trafic et le renvoie à la vraie machine en forgeant des trames Ethernet comportant, cette fois, la véritable adresse MAC.

Cette technique opère au niveau Ethernet, ce qui la rend particulièrement puissante et qui permet de spoofer le trafic IP, voire TCP (si les délais engendrés par la machine pirate sont suffisamment bons).

D'autre part, cela permet de contourner les barrières que constituent les switches en partitionnant le réseau [SEC 02].

Les conséquences vont de la compromission à l'attaque de type Denial Of Service.

3.2.3 Protocole RARP

Le protocole RARP est beaucoup moins utilisé, il signifie *Protocole ARP inversé*, il s'agit donc d'une sorte d'annuaire inversé des adresses logiques et physiques. On est donc en droit de se demander pour quelle raison on aurait besoin de l'adresse physique étant donné que le protocole TCP/IP a besoin uniquement de l'adresse IP pour établir une communication et qu'on la connaît...

En réalité le protocole RARP est essentiellement utilisé pour les stations de travail n'ayant pas de disque dur.

3.2.4 Protocole ICMP

Le Protocole Internet (IP) est utilisé pour la transmission de datagrammes de hôte à hôte à l'intérieur d'un système de réseaux interconnectés. Les appareils raccordant les réseaux entre eux sont appelés des Routeurs. Ces routeurs communiquent entre eux en utilisant différents protocoles afin d'échanger des informations de contrôle et de gestion du réseau. Occasionnellement, un routeur ou un hôte destinataire peut avoir à communiquer vers l'émetteur du datagramme, par exemple, pour signaler une erreur de traitement du datagramme. C'est dans cette perspective qu'a été mis en place le protocole ICMP (Internet Control Message Protocol). Il s'appuie sur le support de base fourni par IP comme s'il s'agissait d'un protocole d'une couche supérieure. ICMP n'en reste pas moins une partie intégrante du protocole IP, et doit de ce fait être implémenté dans chaque module IP.

Les messages ICMP sont envoyés dans diverses situations: par exemple, lorsqu'un datagramme ne peut pas atteindre sa destination, lorsque le routeur manque de réserve de mémoire pour retransmettre correctement le datagramme, ou lorsque le routeur décode de viser l'hôte destinataire via une route alternative pour optimiser le trafic.

Le protocole Internet n'est pas, dans sa définition, absolument fiable. Le but de ces messages de contrôle est de pouvoir signaler l'apparition d'un cas d'erreur dans l'environnement IP, pas de rendre IP fiable. Aucune garantie que le datagramme soit acheminé ni qu'un message de contrôle soit retourné. Certains datagrammes pourront se perdre dans le réseau sans qu'aucun message de contrôle ne le signale. Les protocoles de niveau supérieur s'appuyant sur une couche IP devront implémenter leurs propres mécanismes de contrôle d'erreur et de retransmission si leur objet nécessite un circuit de communication sécurisé.

Les messages ICMP reportent principalement des erreurs concernant le traitement d'un datagramme dans un module IP. Pour éviter de ne pas entrer dans un cercle vicieux de réémission de message de contrôle en réponse à un autre message de contrôle sans fin. aucun message ICMP ne sera réémis en réponse à un message ICMP. De même les messages ICMP ne seront transmis qu'en réponse à un traitement erroné du fragment zéro dans le cas d'un datagramme fragmenter. (Le fragment zéro est celui dont l'offset vaut zéro) [TIL 99].

A) Formats de message

Les messages ICMP sont émis en utilisant l'en-tête IP de base. Le premier octet de la section de données du datagramme est le champ de type ICMP; Sa valeur détermine le format du reste des données dans le datagramme ICMP.

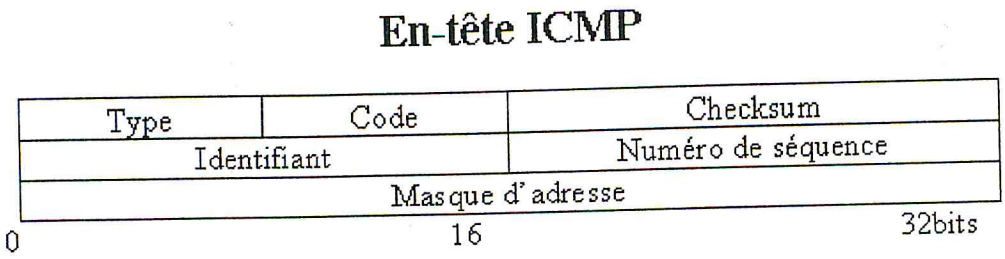


Figure 2. 9: en-tête ICMP [TIL 99]

Type	Code	Message	Signification du message
8	0	Demande d'ECHO	Ce message est utilisé lorsqu'on utilise la commande *PING*. Cette commande, permettant de tester le réseau, envoie un datagramme à un destinataire et lui demande de le restituer
3	0	destinataire inaccessible	Le réseau n'est pas accessible
3	1	destinataire inaccessible	La machine n'est pas accessible
3	2	destinataire inaccessible	Le protocole n'est pas accessible
3	3	destinataire inaccessible	Le port n'est pas accessible
3	4	destinataire inaccessible	Fragmentation nécessaire mais impossible à cause du drapeau (flag) DF
3	5	destinataire inaccessible	Le routage a échoué
3	6	destinataire inaccessible	Réseau inconnu
3	7	destinataire inaccessible	Machine inconnue
3	8	destinataire inaccessible	Machine non connectée au réseau (inutilisé)
3	9	destinataire inaccessible	Communication avec le réseau interdite
3	12	destinataire inaccessible	Machine inaccessible pour ce service
3	11	destinataire inaccessible	Communication interdite (filtrage)
4	0	Source Quench	Le volume de données envoyé est trop important, le routeur envoie ce message pour prévenir qu'il sature afin de demander de réduire la vitesse de transmission
5	0	Redirection pour un hôte	Le routeur remarque que la route d'un ordinateur n'est pas optimale et envoie l'adresse du routeur à rajouter dans la table de routage de l'ordinateur
5	1	Redirection pour un hôte et un service donné	Le routeur remarque que la route d'un ordinateur n'est pas optimale pour un service donné et envoie l'adresse du routeur à rajouter dans la table de routage de l'ordinateur
5	2	Redirection pour un réseau	Le routeur remarque que la route d'un réseau entier n'est pas optimale et envoie l'adresse du routeur à rajouter dans la table de routage des ordinateurs du réseau
5	3	Redirection pour un réseau et un service donné	Le routeur remarque que la route d'un réseau entier n'est pas optimale pour un service donné et envoie l'adresse du routeur à rajouter dans la table de routage des ordinateurs du réseau

Tableau 3 : **Résumé des types de Message** [PIL 03]

B) Les attaques sur ICMP

- ***Ping flooding***, procédé qui consiste à envoyer un nombre important de requête ping vers une cible. Cela provoque, en fonction de l'*OS*, des ralentissements ou un plantage, Smurf c'est un ping flooding particulier ;
- ***Ping of death***, ***sPing***, ***Jolt*** et ***IceNewk***, procédé qui consiste à envoyer un message *ICMP* de type *echo ping* plus grand que sa taille normale. Le pirate le tronque en

multiples paquets que certains systèmes n'arrivent pas à reconstruire et qui se bloquent.

- **Smack–Bloop** : elles consistent à envoyer des messages d'erreur ICMP à l'odinateur cible. Ces attaques provoquent un flood ;
- ***Pong*** ou ***Echo Reply without Request*** ou ***ICMP echo reply attac*k**. Cette méthode envoie la réponse d'un *ping* sans que la victime n'ait envoyé de requête *ping*. Cette attaque permet de déterminer le nombre de machines derrière un routeur ou bien permet de saturer un *routeur*.
- ***Click-WinNewk :*** Cette attaque vise tous les systèmes. Elle consiste à envoyer un message d'erreur ICMP (typiquement, ICMP inaccessible) à l'ordinateur cible ou au serveur auquel la victime est connectée. La victime risque alors d'être déconnectée du réseau ou de serveur. Pour ce protéger il faut configurer le firewall ou le routeurs pour qu'il puise gérer ces messages [SEC 02].

3.2.5 Le routage

Le routage ou acheminement est le ciment permettant d'assurer la cohésion d'Internet. Sans celui-ci, le trafic TCP/IP serait limité à un seul réseau physique. Le routage est la façon de déterminer le trajet optimal des données entre l'émetteur et le récepteur. Le routage est basé sur un algorithme propre au protocole de routage. L'algorithme prend en considération les facteurs les plus importants comme la durée moyenne de transmission, la charge du réseau, la longueur totale du message... Il permet au trafic provenant d'un réseau local d'atteindre sa destination où qu'elle se trouve dans le monde, après avoir probablement traversé plusieurs réseaux intermédiaires. Le rôle décisif que jouent le routage et l'interconnexion complexe des réseaux Internet fait de la conception des protocoles de routage un défi majeur que doivent relever les développeurs de logiciels réseau. Par conséquent, la plupart des études relatives au routage concernent la conception des protocoles ; très peu traitent de la configuration correcte des protocoles de routage. Toutefois, nombre de problèmes quotidiens résultent plutôt d'une mauvaise configuration des routeurs utilisés que de l'emploi d'algorithmes mal conçus. C'est le rôle de l'administrateur système de s'assurer que la configuration du routage est correcte.

A) Configurations de routage

Il convient d'établir une distinction entre le routage proprement dit et les protocoles de routage.

La configuration du routage d'un réseau spécifique ne requiert pas toujours l'utilisation d'un protocole de routage. Dans les situations où les informations de routage ne subissent aucune modification (lorsqu'il n'existe qu'une seule route) l'administrateur système crée généralement une table de routage manuellement.

Les trois configurations de routage les plus courantes sont :

1. Routage minimal : Un réseau complètement isolé des autres réseaux TCP/IP requiert uniquement un acheminement minimal. Une fois l'interface configurée, le système crée une table de routage minimale. Si votre réseau ne peut accéder aux réseaux TCP/IP directement et si on ne crée pas de sous-réseaux, l'acheminement minimal peut s'avérer la seule et unique table de routage nécessaire.

2. Routage statique : Un réseau doté d'un nombre limité de passerelles vers d'autres réseaux TCP/IP peut être configuré en ayant recours à l'acheminement statique. L'administrateur système crée manuellement une table de routage statique au moyen de la commande route. Les tables de routage statiques ne s'adaptent pas aux modifications apportées au réseau et pourtant doivent être utilisées uniquement sur les réseaux dont les routes ne subissent aucune modification. Toutefois, si on peut atteindre des destinations uniquement au travers d'une seule route, une route statique constitue alors le meilleur choix.

3. Routage Dynamique : Un réseau pouvant utiliser plusieurs routes vers la même destination doit utiliser l'acheminement dynamique. Une table de routage dynamique est créée à partir des informations échangées par les protocoles de routage. Les protocoles sont conçus pour distribuer les informations qui permettent d'adapter dynamiquement les routes afin de refléter les modifications apportées aux conditions de fonctionnement du réseau. Les protocoles de routage prennent en charge des situations complexes d'acheminement plus rapidement et avec plus de précision que l'administrateur système. Les protocoles de routage ne sont pas conçus uniquement pour commuter vers une route de secours lorsque la voie principale est inutilisable ; ils sont aussi conçus pour déterminer quelle est la "meilleure" route vers une destination donnée. Sur un réseau disposant de plusieurs chemins vers la même destination, il convient d'utiliser un protocole de routage dynamique [PIL 03].

Adresse de destination	Adresse du prochain routeur directement accessible	Interface
194.56.32.124	131.124.51.108	2
110.78.202.15	131.124.51.108	2
53.114.24.239	194.8.212.6	3
187.218.176.54	129.15.64.87	1

Tableau 4 : une table de routage [PIL 03]

3.2.6 Diversité des protocoles de routage

Tous les protocoles de routage exécutent les mêmes fonctions de base. Ils déterminent la "meilleure" route vers chaque destination et distribuent les informations d'acheminement entre les systèmes d'un réseau. Dans la terminologie TCP/IP, ces systèmes de réseaux indépendants s'appellent des systèmes autonomes. Les modalités d'exécution de ces fonctions, en particulier les procédures de sélection des meilleures routes permettent de distinguer les différents protocoles. Les protocoles de routage se classent en deux groupes généraux : protocoles internes et externes.

Système autonome : Il s'agit d'un ensemble de routeurs dépendant d'une seule administration technique, ces routeurs utilisent le protocole de passerelle interne ainsi que les métriques courantes pour acheminer les paquets au sein de l'AS (Autonomous

System). Ils utilisent aussi le protocole de passerelle externe pour acheminer les paquets vers d'autres AS... Les autres AS considèrent que l'administration d'un AS déterminé dispose d'un plan d'acheminement intérieur cohérent présentant une vue d'ensemble homogène des réseaux accessibles à travers cette AS. Du point de vue de l'acheminement externe, l'AS peut être considéré comme monolithique...Les protocoles de routage externe se chargent d'introduire des informations d'acheminement dans ces monolithes et d'en extraire [RFC 1163]. Les deux sections suivantes fournissent une vue d'ensemble des protocoles de routage actuellement utilisés.

A) Protocoles de routage interne

Un protocole interne est un protocole de routage utilisé au sein (interne à) d'un système de réseaux indépendants. Au sein d'un système autonome (AS), les informations d'acheminement sont échangées au moyen d'un protocole interne choisi par l'administration du système autonome. Il existe plusieurs protocoles internes :

A.1 Le protocole RIP

RIP signifie *Routing Information Protocol* (protocole d'information de routage). Il s'agit d'un protocole de type *Vector Distance* (Vecteur Distance), c'est à dire que chaque routeur communique aux autres routeurs la distance qui les sépare (le nombre de saut qui les sépare). Ainsi, lorsqu'un routeur reçoit un de ces messages il incrémente cette distance de 1 et communique le message aux routeurs directement accessibles. Les routeurs peuvent donc conserver de cette façon la route optimale d'un message en stockant l'adresse du routeur suivant dans la table de routage de telle façon que le nombre de saut pour atteindre un réseau soit minimal. Toutefois ce protocole ne prend en compte que la distance entre deux routeurs en termes de saut, mais il ne considère pas l'état de la liaison afin de choisir la meilleure bande passante possible [PIL 03].

A.2 Le protocole OSPF

OSPF (*Open Shortest Path First*) est plus performant que RIP. Il s'agit d'un protocole de type *protocole route-link* (*Protocole d'état des liens*), cela signifie que, contrairement à RIP, ce protocole n'envoie pas aux routeurs adjacents le nombre de sauts qui les sépare, mais l'état de la liaison qui les sépare. De cette façon, chaque routeur est capable de dresser une carte de l'état du réseau et peut par conséquent choisir à tout moment la route la plus appropriée pour un message donné. De plus, ce routeur évite aux routeurs intermédiaires d'avoir à incrémenter le nombre de sauts, ce qui se traduit par une information beaucoup moins abondante, ce qui permet d'avoir une meilleure bande passante utile qu'avec RIP [PIL 03].

B. Protocoles de routage externes

Les protocoles de routage externes sont utilisés pour échanger des informations d'acheminement entre systèmes autonomes. Les informations d'acheminement transférées entre des systèmes autonomes s'appellent informations d'accessibilité. Les

informations d'accessibilité correspondent simplement à des informations concernant les réseaux accessibles à travers un système autonome spécifique.

B.1 Protocole EGP

Lors de la conception d'EGP (Externe Gateway protocole), le réseau dépendait d'un groupe de passerelles noyau à deux ou plusieurs niveaux de sécurité pour traiter et distribuer les routes provenant de l'ensemble des systèmes autonomes. Ces passerelles noyau disposaient alors des informations nécessaires pour choisir les meilleures voies externes. Les informations d'accessibilité EGP ont été transmises aux passerelles noyau, au sein desquelles les informations ont été groupées et retransmises ensuite aux systèmes autonomes. Etant donné que le nombre de systèmes autonomes et de réseaux connectés à Internet ne cesse d'augmenter, les passerelles noyau se trouvent dans l'impossibilité de traiter cette charge de travail herculéenne. C'est la raison pour laquelle Internet passe progressivement à une architecture plus répartie qui distribue la charge inhérente à la prise en charge des routes entre systèmes autonomes.

Dans une architecture répartie, les systèmes autonomes requièrent l'utilisation de protocoles de routage, internes et externes, pouvant effectuer des choix intelligents d'acheminement.

Contrairement aux protocoles internes décrits précédemment, EGP met à jour les informations de l'algorithme du vecteur distance, mais n'évalue pas ces informations. Les valeurs de l'algorithme vecteur de distance provenant de différents systèmes autonomes ne peuvent pas être comparées directement puisque chaque AS utilise éventuellement différents critères pour déterminer ces valeurs. Par conséquent, EGP délègue le choix de la meilleure route à quelqu'un d'autre. EGP est un protocole permettant d'échanger les informations d'acheminement avec les passerelles appartenant à d'autres systèmes autonomes. Toutefois, avant qu'un système n'envoie des informations d'acheminement à une passerelle distante, celui-ci doit d'abord échanger les messages Hello et I-Heard-You du protocole EGP avec cette passerelle. Les messages Hello et I-Heard-You sont des paquets EGP spéciaux utilisés pour établir un dialogue entre deux passerelles utilisant EGP. Les ordinateurs communiquant via EGP s'appellent voisins EGP (EGP neighbours) et l'échange des messages Hello et I-H-U s'appelle acquisition d'un voisin (acquiring a neighbour). Dès qu'un voisin est acquis, le système demande au voisin de lui transmettre les informations d'acheminement. La demande de transmission de ces informations s'appelle un message d'interrogation. Le voisin répond en envoyant un paquet contenant les informations d'accessibilité appelées « mise à jour ». Si le système reçoit un message d'interrogation provenant de son voisin EGP, il lui renvoie son propre paquet de mise à jour. Lorsque le système reçoit une mise à jour provenant de son voisin, il inclut les routes de mise à jour dans sa table de routage local. Si le voisin ne répond pas à trois messages d'interrogation consécutifs, le système considère que le

voisin est en panne et supprime les routes du voisin figurant dans sa table. Une passerelle exécutant EGP annonce qu'elle peut atteindre les réseaux intégrés à ce système autonome. Excepté pour un petit sous-ensemble de passerelles s'exécutant comme des passerelles noyau, une passerelle EGP n'annonce pas qu'elle peut atteindre des réseaux externes à son système autonome.

B.2 Protocole BGP

Un nouveau protocole de routage externe, Protocole de passerelle de limite (BGP : Border Gateway Protocol), commence à remplacer EGP.

Comme EGP, BGP échange des informations d'accessibilité entre des systèmes autonomes. Toutefois, BGP assure un nombre plus important de fonctions :

BGP peut fournir des informations plus détaillées concernant chaque route et peut les utiliser pour sélectionner la meilleure voie. BGP appelle ces informations, attributs de chemin. Les attributs peuvent inclure des informations utilisées pour sélectionner des routes en fonctions des préférences administratives. Ce type d'acheminement (appelé parfois acheminement en fonction de la politique) utilise des raisons non techniques (par exemple des considérations politiques, organisationnelles ou de sécurité) pour prendre les décisions d'acheminement. Par conséquent, BGP améliore les possibilités du système quant à la sélection des routes et à la mise en œuvre des politiques d'acheminement. Ces caractéristiques s'avèrent importantes pour les réseaux ne dépendant pas des passerelles noyau pour l'exécution de ces tâches. Les possibilités de BGP permettent de mettre en œuvre une nouvelle structure de réseau constituée de systèmes autonomes équivalents pouvant évoluer davantage que l'ancienne structure hiérarchique.

3.3 LA COUCHE TRANSPORT

Les protocoles des couches précédentes permettaient d'envoyer des informations d'une machine à une autre. La couche transport permet à des applications tournant sur des machines distantes de communiquer. Le problème consiste à identifier ces applications. En effet, suivant la machine et son système d'exploitation, l'application pourra être un programme, une tâche, un processus... De plus, la dénomination de l'application peut varier d'un système à un autre, c'est la raison pour laquelle un système de numéro a été mis en place afin de pouvoir associer un type d'application à un type de données, ces identifiants sont appelés ports,

3.3.1 Notion de Port

La plupart des systèmes d'exploitation étant multiprogrammes, les processus s'y exécutent simultanément et ils y sont crées et détruits dynamiquement. Ceci peut prêter confusion car les émetteurs sont rarement informés et les messages peuvent donc être perdus. Pour cela le protocole TCP/IP fournie sur chaque machine un ensemble de destinations abstraites appelées «ports de protocole », chaque port est

identifié par un entier positif et le système d'exploitation local fournit les mécanismes d'interface que les processus utilisent pour spécifier un port et y accéder.
Ce mécanisme permet aux processus émetteurs d'identifier les destinations selon les fonctions qu'elles assurent et de masquer l'identité du processus destination qui en cas de destruction peut être remplacé par un autre.
Les systèmes d'exploitation assurent un accès synchrone aux ports, c'est à dire : si un processus tente d'accéder à un port avant que des données ne soient arrivées, il est bloqué jusqu'à ce que les données arrivent. D'autre part si un datagramme arrive sur un port il est stocké dans une file d'attente jusqu'à son extraction par un processus.
Pour établir une communication entre deux machines, l'émetteur doit connaître l'adresse IP de la machine destination et le numéro de port associer au protocole sur cette machine.
Chaque message doit contenir le numéro de PORT source et le numéro de port destination ports.
La couche transport contient deux protocoles permettant à deux applications d'échanger des données indépendamment du type de réseau emprunté (c'est à dire indépendamment des couches inférieures...), il s'agit des protocoles suivants:

- TCP, un protocole orienté connexion qui assure le contrôle des erreurs
- UDP, un protocole non orienté connexion dont le contrôle d'erreur est négligé

3.3.2 Protocole UDP

Le protocole User Datagramme Protocol (UDP) est défini dans le but de fournir une communication par paquet unique entre deux processus dans un environnement réseau étendu. Il suppose l'utilisation du protocole IP comme support de base à la communication. Il définit une procédure permettant à une application d'envoyer un message court à une autre application, selon un mécanisme minimalisée.
Il est transactionnel, et ne garantit ni la délivrance du message, ni son éventuelle duplication. Les applications nécessitant une transmission fiabilisée et ordonnée d'un flux de données utiliseront de préférence le protocole TCP.

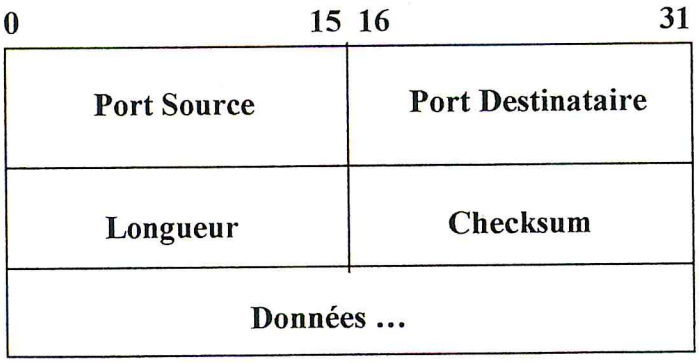


Figure 2. 10: Format du paquet UDP [TIL 99]

- **Le Port Source :** indique le numéro de port du processus émetteur, et l'on supposera, en l'absence d'informations complémentaires, que toute réponse devra y être dirigée. S'il n'est pas utilisé, ce champ conservera une valeur 0 ;
- **Le Port Destinataire :** a une signification dans le cadre d'adresses Internet particulières ;
- **La Longueur :** compte le nombre d'octets dans le datagramme entier y compris le présent en-tête. (Et par conséquent la longueur minimale mentionnée dans ce champ vaut huit, si le datagramme ne transporte aucune donnée) ;
- **Le Checksum :** se calcule en prenant le complément à un de la somme sur 16 bits des compléments à un calculé sur un pseudo en-tête constitué de l'information typique d'une en-tête IP, l'en-tête UDP elle-même, et les données, le tout additionné d'un octet nul éventuel afin que le nombre total d'octets soit pair.

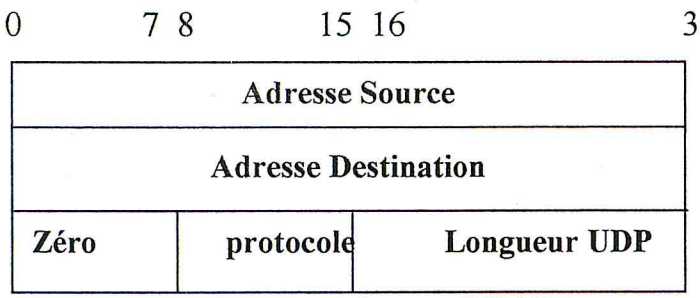


Figure 2. 11: **Pseudo entête UDP** [TIL 99]

Le pré en-tête ajouté avant l'en-tête UDP contient l'adresse IP source, l'adresse IP destinataire, le code de protocole, et la longueur du segment UDP. Cette information permet d'augmenter l'immunité du réseau aux erreurs de routage de datagramme. La procédure de calcul du Checksum est la même que pour TCP.

Si le calcul du checksum vaut zéro, il sera transmis tous ses bits à un (le complément à un). Un Checksum transmis avec une valeur zéro a effectivement une signification particulière. Dans ce cas, le segment indique qu'aucun Checksum n'a été calculé (pour des besoins de mise au point ou pour des protocoles de niveaux supérieurs qui rendent cette vérification inutile) [TIL 99].

A) Interface Utilisateur

L'interface utilisateur doit permettre l'ouverture de nouveaux ports de réception, la réception des données et leur transmission ainsi que celle de l'adresse source à l'application sur le port de réception mis en place, et doit mettre en place une commande permettant l'émission d'un datagramme, par laquelle seront spécifiés les données, l'adresse et ports source et destination à utiliser.

B) Interface IP

Le module UDP doit extraire les adresses source et destination de l'en-tête IP, et vérifier le numéro de protocole. Une interface UDP/IP plausible pourrait retourner le datagramme entier y compris l'en-tête Internet en réponse du datagramme reçu. Une interface devra pour cela permettre à UDP de passer un datagramme Internet complet

avec une en-tête IP à la couche IP elle même pour émission. IP n'aura plus qu'à vérifier la cohérence des champs d'en-tête IP préparés par UDP et calculer le Checksum.

c) attaque sur UDP

-Attaque Bionk : cette attaque vise les systèmes Win32. Elle consiste à envoyer des packets UDP corrompus sur tous les ports ouverts. L'ordinateur victime ne gère pas ces paquets et provoque un plantage. Pour ce protéger il recommander d'utiliser un firewall pour refuser les packets UDP corrompus [SEC 02] ;

-Attaque Bonk : vise les système WinNT 3.51 et 4.0. Elle consiste à envoyer des packets UDP corrompus sur le port 53. Chaque packet UDP corrumpu est constitué de deux fragements IP assemblés en un UDP. Les offsets qui se superposent ont pour consécuence de faire écraser la seconde moitié de l'entête UDP par le second packet IP. L'ordinateur victime ne gère pas ces paquets et provoque un plantage (message STOP 0x00000000A) dû à une allocation excessive de la mémoire du noyau ;

-Attaque Snork : vise les systèmes WinNT. Elle consiste à envoyer une trame UDP provenant du port 7(Echo), 19 (Chargen) ou 135, et ayant pour destination le port 135 (Microsoft Location Service). Si les services sont lancés, cela a pour conséquence d'établir une communication de durée infinie, et génère des trames non nécessaires. Cela réduit considérablement la bande passante et la puissance CPU [SEC 02] ;

***-UDP* 0** : vise à envoyer un paquet *UDP* sur le port 0. Cette action peut planter l'ordinateur.

3.3.3 Protocole TCP

Le protocole TCP (Transmission Contrôle Protocole) est un protocole sécurisé orienté connexion conçu pour s'implanter dans un ensemble de protocoles multicouches, supportant le fonctionnement de réseaux hétérogènes. TCP fournit un moyen d'établir une communication fiable entre deux tâches exécutées sur deux ordinateurs autonomes raccordés à un réseau de données. Le protocole TCP s'affranchit le plus possible de la fiabilité intrinsèque des couches inférieures de communication sur lesquelles il s'appuie. TCP suppose donc uniquement que les couches de communication qui lui sont inférieures lui procurent un service de transmission de paquet simple, dont la qualité n'est pas garantie. En principe, TCP doit pouvoir supporter la transmission de données sur une large gamme d'implémentations de réseaux, depuis les liaisons filaires câblées, jusqu'aux réseaux commutés, ou asynchrones.

TCP s'intègre dans une architecture multicouche des protocoles, juste au-dessus du protocole Internet IP. Ce dernier permet à TCP l'envoi et la réception de segments de longueur variable, encapsulés dans datagramme. Le datagramme Internet dispose des mécanismes permettant l'adressage d'un service TCP source et un destinataire, quelles que soient leur position dans le réseau. Le protocole IP s'occupe aussi de la

fragmentation et du réassemblage des paquets TCP lors de la traversée de réseaux de plus faibles caractéristiques.

0	4	10	16	24	31
Port Source			Port Destination		
Numéro de séquence					
Numéro d'accusé de réception					
Long-Entête	Réservé	Bits de code	Fenêtre		
Checksum			Pointeur de données urgentes		
Options				Padding	
Données					

Figure 2. 12: Format du paquet TCP [TIL 99].

- **Port source** (16 bits) : Le numéro de port de la source ;
- **Port Destinataire** (16 bits) : Le numéro de port du destinataire ;
- **Numéro de séquence** (32 bits) : Le numéro du premier octet de données par rapport au début de la transmission (sauf si SYN est marqué). Si SYN est marqué, le numéro de séquence est le numéro de séquence initial (ISN) et le premier octet à pour numéro ISN+1 ;
- **Accusé de réception** (32 bits) : Si ACK est marqué ce champ contient le numéro de séquence du prochain octet que le récepteur s'attend à recevoir. Une fois la connexion établie, ce champ est toujours renseignée ;
- **Long Entête** (4 bits) : La taille de l'en-tête TCP en nombre de mots de 32 bits. Il indique là ou commence les données. L'en-tête TCP, dans tous les cas a une taille correspondant à un nombre entier de mots de 32 bits ;
- **Réservé** (6 bits) : Réservés pour usage futur. Doivent nécessairement être à 0;
- **Bits de contrôle** (6 bits de gauche à droite) :
  - **URG :** Pointeur de données urgentes significatif
  - **ACK :** Accusé de réception significatif
  - **PSH :** Fonction Push
  - **RST :** Réinitialisation de la connexion
  - **SYN :** Synchronisation des numéros de séquence
  - **FIN :** Fin de transmission ;
- **Fenêtre** (16 bits) : Le nombre d'octets à partir de la position marquée dans l'accusé de réception que le récepteur est capable de recevoir ;
- **Checksum** (16 bits) : Le Checksum est constitué en calculant le complément à 1 sur 16 bits de la somme des compléments à 1 des octets de l'en-tête et des données pris deux par deux (mots de 16 bits).Le Checksum couvre de plus une pseudo entête de 96 bits préfixée à l'en-tête TCP :

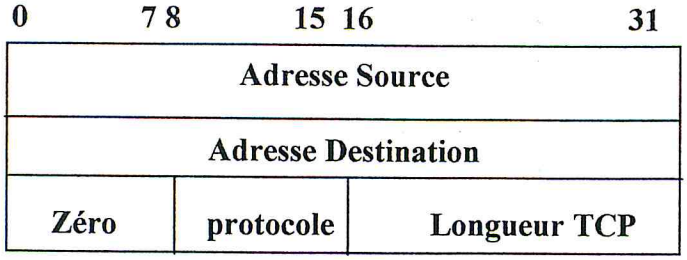


Figure 2. 13: Pseudo entête TCP [TIL 99]

- **Pointeur de données urgentes** (16 bits) : Communique la position d'une donnée urgente en donnant son décalage par rapport au numéro de séquence. Le pointeur doit pointer sur l'octet suivant la donnée urgente. Ce champ n'est interprété que lorsque URG est marqué ;
- **Options** (variable) : Les champs d'option peuvent occuper un espace de taille variable à la fin de l'en-tête TCP, ils formeront toujours un multiple de 8 bits. Toutes les options sont prises en compte par le Checksum. Un paramètre d'option commence toujours sur un nouvel octet. Il est défini deux formats types pour les options :
  - Cas 1 : Option mono-octet,
  - Cas 2 : Octet de type d'option, octet de longueur d'option, octets de valeurs d'option ;
- **Bourrage** (padding) (variable) : Les octets de bourrage terminent l'en-tête TCP :
  - de sorte que le nombre d'octet de celle-ci soit toujours multiple de 4 (32 bits)
  - de sorte que l'offset de données marquer dans l'en-tête corresponde bien au début des données applicatives.

A) Interfaces utilisateur

TCP s'interface avec un processus utilisateur ou applicatif et un protocole de niveau inférieur du type Internet Protocol.

L'interface avec les applicatifs consiste en un ensemble de commandes comme le ferait une application à un système d'exploitation pour la manipulation de fichiers. Par exemple, on trouvera des commandes pour établir et rompre une communication, pour envoyer ou recevoir des données sur une connexion ouverte. Il est aussi prévu que TCP puisse communiquer avec les applications sur un mode asynchrone. Bien qu'une grande liberté soit laissé aux développeurs pour la construction d'interfaces TCP pour un environnement donné, des fonctionnalités minimales sont requises pour reconnaître la validité TCP de l'implémentation.

B) Interface avec les couches inférieures

L'interface entre TCP et les protocoles de couche base restent largement non spécifiés excepté le fait qu'il doit y exister un mécanisme de transfert asynchrone de données. En général, c'est le protocole inférieur qui est sensé fournir la définition de cette

interface. TCP assume un fonctionnement avec un large ensemble de protocoles réseau. Dans ce document, nous nous limiterons au fonctionnement avec IP.

C) Fiabilité des transferts

Le protocole TCP permet d'assurer le transfert des données de façon fiable, bien qu'il utilise le protocole IP, qui n'intègre aucun contrôle de livraison de datagramme.

En réalité, le protocole TCP possède un système d'accusé de réception permettant au client et au serveur de s'assurer de la bonne réception mutuelle des données. Lors de l'émission d'un segment, un numéro d'ordre (appelé aussi numéro de séquence) est associé. A la réception d'un segment de donnée, la machine réceptrice va retourner un segment de donnée dont le drapeau ACK est à 1 (afin de signaler qu'il s'agit d'un accusé de réception) accompagné d'un numéro d'accusé de réception égal au numéro de séquence précédent.

De plus, grâce à une minuterie déclenchée dès réception d'un segment au niveau de la machine émettrice, le segment est réexpédié dès que le temps imparti est écoulé, car dans ce cas la machine émettrice considère que le segment est perdu...

Toutefois, si le segment n'est pas perdu et qu'il arrive tout de même à destination, la machine réceptrice saura grâce au numéro de séquence qu'il s'agit d'un doublon et ne conservera que le premier segment arrivé à destination...

D) Etablissement d'une connexion

Etant donné que ce processus de communication, qui se fait grâce à une émission de données et d'un accusé de réception, est basé sur un numéro d'ordre (appelé généralement numéro de séquence), il faut que les machines émettrices et réceptrices (client et serveur) connaissent le numéro de séquence initial de l'autre machine.

L'établissement de la connexion entre deux applications se fait souvent selon le schéma suivant :

- Les ports TCP doivent être ouverts,
- L'application sur le serveur est passive, c'est à dire que l'application est à l'écoute, en attente d'une connexion,
- L'application sur le client fait une requête de connexion sur le serveur dont l'application est en ouverture passive. L'application du client est dite "en ouverture passive",

Les deux machines doivent donc synchroniser leurs séquences grâce à un mécanisme communément appelé *three ways handshake* (*poignée de main en trois temps*), que l'on retrouve aussi lors de la clôture de session.

Ce dialogue permet d'initier la communication, il se déroule en trois temps, comme sa dénomination l'indique :

- Dans un premier temps la machine émettrice (le client) transmet un segment dont le drapeau SYN est à 1 (pour signaler qu'il s'agit d'un segment de synchronisation), avec un numéro d'ordre N, que l'on appelle numéro d'ordre initial du client ;

- Dans un second temps la machine réceptrice (le serveur) reçoit le segment initial provenant du client, puis lui envoie un accusé de réception, c'est à dire un segment dont le drapeau ACK est à 1 et le drapeau SYN est à 1 (car il s'agit là encore d'une synchronisation). Ce segment contient le numéro d'ordre de cette machine (du serveur) qui est le numéro d'ordre initial du client. Le champ le plus important de ce segment est le champ accusé de réception qui contient le numéro d'ordre initial du client, incrémenté de 1 ;
- Enfin, le client transmet au serveur un accusé de réception, c'est à dire un segment dont le drapeau ACK est à 1, dont le drapeau SYN est à zéro (il ne s'agit plus d'un segment de synchronisation). Son numéro d'ordre est incrémenté et le numéro d'accusé de réception représente le numéro de séquence initial du serveur incrémenté de 1.

Suite à cette séquence comportant trois échanges les deux machines sont synchronisées et la communication peut commencer.

E) attaque sur TCP

-Spoofing :

Il existe une technique de piratage, appelée spoofing, permettant de corrompre cette relation d'approbation à des fins malicieuses.

Le client peut demander à mettre fin à une connexion au même titre que le serveur. La fin de la connexion se fait de la manière suivante :

- Une des machines envoie un segment avec le drapeau *FIN* à 1, et l'application se met en état d'attente de fin, c'est à dire qu'elle finit de recevoir le segment en cours et ignore les suivants ;
- Après réception de ce segment, l'autre machine envoie un accusé de réception avec le drapeau *FIN* à 1 et continue d'expédier les segments en cours. Suite à cela la machine informe l'application qu'un segment *FIN* a été reçu, puis envoie un segment *FIN* à l'autre machine, ce qui clôture la connexion.

- **Tear Drop :** elle consiste à envoyer des packets TCP qui se recouvrent. Lorsque l'ordinateur victime reçoit ces packets, il tente de les reconstruire. N'y arrivant pas, cela provoque un plantage ;

-TCP-SYN/Flooding : Comme on a déjà expliqué, quand un système client essaie d'établir une connexion TCP à un système fournissant un service (le serveur), le client et le serveur échange une séquence de messages.

Les abus viennent au moment où le serveur a renvoyé un accusé de réception du SYN (ACK-SYN) au client mais n'a pas reçu le «ACK» du client. C'est alors une connexion à demi-ouverte. Le serveur construit dans sa mémoire système une structure de données décrivant toutes les connexions courantes. Cette structure de données est

de taille finie, ce qui veut dire qu'il peut se créer un dépassement de capacité en créant intentionnellement trop de connexions partiellement ouvertes.
L'ordinateur de l'agresseur envoie des messages SYN à la machine victime; ceux-ci paraissent provenir d'un ordinateur bien défini mais qui en fait, fait référence à un système client qui n'est pas capable de répondre au message SYN-ACK. Ce qui veut dire que le message ACK de confirmation finale ne sera jamais renvoyé au serveur victime. Normalement, il y a un système de « time-out » (i.e. si le système attend un événement particulier, au bout d'un certains temps, il considère que cet événement n'apparaîtra plus et génère une erreur ou un message) associé à chaque connexion ouverte, donc les demi-connexions devraient expirer et le serveur victime ainsi récupérer de la place libre dans sa mémoire pour d'autres connexions. Toutefois, le système agresseur continue d'envoyer des paquets plus vite que le temps nécessaire au serveur pour faire expirer les demi-connexions. La localisation de l'attaque est très complexe car les adresses contenues dans les paquets SYN envoyés sont très souvent falsifiées. Il n'y a donc pas de moyen de déterminer sa véritable source. Internet faisant suivre les paquets grâce à l'adresse de destination, le seul moyen s'affranchir de ces attaques est de valider la source d'un paquet en utilisant le filtrage.

F) Le multiplexage

TCP permet d'effectuer une tâche importante: le multiplexage/démultiplexage, c'est à dire faire transiter sur une même ligne des données provenant d'applications diverses ou en d'autres mots mettre en série des informations arrivant en parallèle.
Ces opérations sont réalisées grâce au concept de ports (ou sockets), c'est à dire un numéro associé à un type d'application, qui, combiné à une adresse IP, permet de déterminer de façon unique une application qui tourne sur une machine donnée.

G) Le contrôle de flux

Dans de nombreux cas, il est possible de limiter le nombre d'accusés de réception, afin de désengorger le réseau, en fixant un nombre de séquence au bout duquel un accusé de réception est nécessaire. Ce nombre est en fait stocké dans le champ *fenêtre* de l'en-tête TCP/IP. On appelle effectivement cette méthode **"méthode de la fenêtre glissante"** car on définit en quelque sorte une fourchette de séquences n'ayant pas besoin d'accusé de réception, et celle-ci se déplace au fur et à mesure que les accusés de réception sont reçus. De plus, la taille de cette fenêtre n'est pas fixe. En effet, le serveur peut inclure dans ses accusés de réception en stockant dans le champ fenêtre la taille de la fenêtre qui lui semble la plus adaptée. Ainsi, lorsque l'accusé de réception indique une demande d'augmentation de la fenêtre, le client va déplacer le bord droit de la fenêtre.
Par contre, dans le cas d'une diminution, le client ne va pas déplacer le bord droit de la fenêtre vers la gauche mais attendre que le bord gauche avance (avec l'arrivée des accusés de réception).

3.3.4 ADRESSE DES APPLICATIONS

Pour effectuer l'adresse des applications, TCP et UDP gèrent des Ports. Les numéros de port inférieurs à 512, ce sont les ports réservés. Les numéros de port supérieurs à 1024 sont disponibles aux utilisateurs. La combinaison d'une adresse IP et d'un numéro de port est appelée une « prise » (***socket***), elle identifier d'une façon unique une entité d'application s'exécutant dans une interconnexion de réseaux.
Evidemment, la notion d'adressage des applications constitue un exemple du multiplexage des protocoles :

- Au niveau de la couche interface, chaque réseau physique distingue habituellement ses clients (entités au niveau de la couche Internet), grâce aux différentes valeurs du champ ***type*** de l'entête Ethernet (ainsi, **Ethernet** utilise la valeur 0x0800 pour designer **IP**).
- Au niveau de la couche Internet, IP distingue les clients à l'aide de différente valeurs du champ **protocole** de l'entête IP (ainsi, **IP** utilise la valeur 17 pour designer le protocole **UDP**).
- Au niveau de la couche transport, TCP et UDP distinguent les clients (les entités de la couche application) au moyen des ***numéro de ports*** différents (ex : **UDP** utilise la valeur décimale 161 pour designer **SNMP**).

En fin ce tableau présente les grandes différences entre UDP et TCP :

	Protocole UDP	Protocole TCP
Reprise sur erreur	**Non**	**Oui**
Contrôle de flux	**Non**	**Oui**
Connexion	**Sans**	**Avec**
Surcharge du réseau	**Non**	**Oui**
Fiabilité	**Non**	**Oui**

Tableau 5 : Tableau comparatif entre les TCP et UDP [TIL 99]

3.4 LA COUCHE APPLICATION

La couche application est la couche située au sommet des couches de protocoles TCP/IP. Celle-ci contient les applications réseaux permettant de communiquer grâce aux couches inférieures. Les applications de cette couche sont de différents types, mais la plupart sont des services réseau, c'est à dire des applications fournies à l'utilisateur pour assurer l'interface avec le système d'exploitation. On peut les classer selon les services qu'ils rendent :

- Les services de gestion (transfert) de fichier et d'impression
- Les services de connexion au réseau
- Les services de connexion à distance
- Les utilitaires Internet divers

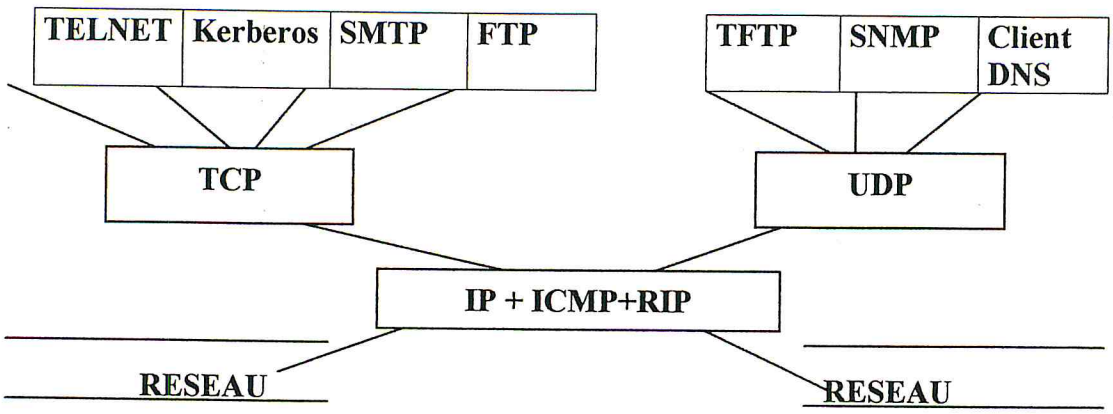


Figure 2. 14: L'architecture TCP/IP [TIL 99]

3.4.1 PROTOCOLE HTTP

Le protocole HTTP (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990. La version 0.9 était uniquement destinée à transférer des données sur Internet. La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MIME. Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML) localisé grâce à une chaîne de caractères appelée URL entre un navigateur (le client) et un serveur Web (appelé d'ailleurs Httpd) La communication entre le navigateur et le serveur se fait en deux temps:

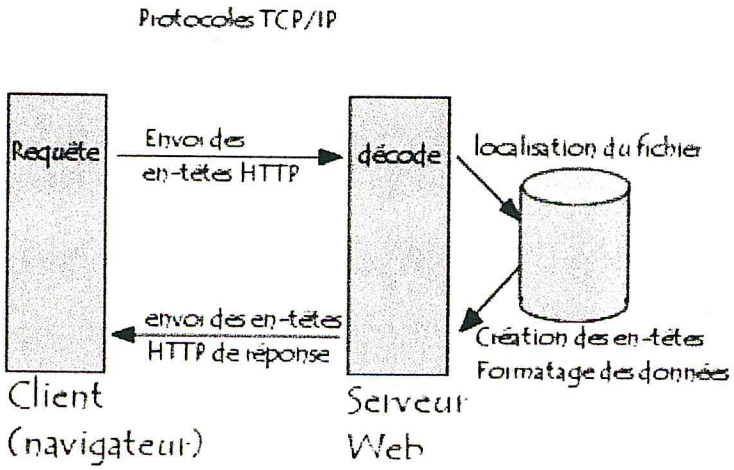


Figure 2. 15 **: Communication entre navigateur et serveur** [PIL 03]

- Le navigateur effectue une requête HTTP
- Le serveur traite la requête puis envoie une réponse HTTP

En réalité la communication s'effectue en plus de temps si on considère le traitement de la requête par le serveur.

A) Requête http

Une requête HTTP est un ensemble de lignes envoyé au serveur par le navigateur. Elle comprend :

-Une ligne de requête : c'est une ligne précisant le type de document demandé, la méthode qui doit être appliqué, et la version du protocole utilisée. La ligne comprend trois éléments devant être séparé par un espace : La méthode, l'URL et la version du protocole utilisé par le client (généralement *HTTP/1.0*)

-Les champs d'en-tête de la requête : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client (Navigateur, système d'exploitation,...). Chacune de ces lignes est composé d'un nom qualifiant le type d'en-tête, suivi de deux points (:) et de la valeur de l'en-tête ;

-Le corps de la requête: C'est un ensemble de ligne optionnel devant être séparé des lignes précédentes par une ligne vide et permettant par exemple un envoi de données par une commande POST lors de l'envoi de données au serveur par un formulaire.

Commande	Description
GET	**Requête de la ressource située à l'URL spécifié**
HEAD	**Requête de la ressource située à l'URL spécifié**
POST	**Envoi de données au programme situé à l'URL spécifié**
PUT	**Envoi de données à l'URL spécifié**
DELETE	**Suppression de la ressource située à l'URL spécifié**

Tableau 6 : les commandes http [PIL 03]

B) Risques du protocole http

Les risques qui peuvent être engendré via HTTP se posent à deux niveaux :

- d'un client malveillant vers un serveur HTTP,
- d'un serveur HTTP malveillant à un client HTTP.

Un attaquant est susceptible de faire subir à un serveur HTTP ce qu'il ferait subir à n'importe quel serveur, qu'il s'agisse d'accéder à des données privées, ou d'obtenir des droits particulier. Une restriction des données présentes sur le serveur et une bonne configuration va permettre de limiter les risques de ce type. Il existe cependant une caractéristique spéciale des serveurs HTTP : ceux-ci utilisent souvent des programmes externes notamment des scripts CGI (Common Gateway Interface). Pour ces programmes, un agresseur va vouloir les tromper pour les utiliser à ses fins. Il peut également vouloir installer ses propres programmes et les faire exécuter sur le serveur. Dans ces deux cas, on conseille de restreindre au maximum les zones d'accès au serveur, et de fournir un environnement sûr.

Les problèmes de sécurité des clients HTTP sont plus complexes que pour les serveurs, essentiellement parce que ces clients ont été conçu pour être extensibles et pour lancer des programmes externes, dont le rôle sera de traiter certains types de données (comme faire du streaming vidéo).

On encore des attaques de type DoS qui vise le serveur comme la suivante :

Bad HTTP request, vise tous les systèmes et consiste à envoyer des requêtes HTTP mal formaté. Le serveur peut alors planté.

Un agresseur va tenter de tirer avantages de ces fonctionnalités, ou bien de faire en sorte que l'utilisateur du programme client modifie sa configuration ou ajoute des programmes spécifiques (fournis par l'agresseur sous une fausse raison), tout cela dans le but d'utiliser le client HTTP comme l'agresseur le souhaite.

3.4.2 LE PROTOCOLE SMTP

Simple Mail Transfer Protocol, (*Protocole Simple de Transfert de Courrier)* est le 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 courrier est remis directement au serveur de courrier du destinataire. Le protocole SMTP fonctionne grâce à des commandes textuelles envoyées au serveur SMTP (par défaut sur le port 25). Chacune des commandes envoyées par le client (validée par un appui sur la touche entrée) est suivi d'une réponse du serveur SMTP composée d'un numéro et d'un message descriptif.
Voici un scénario de demande d'envoi de mail à un serveur SMTP

- Lors de l'ouverture de la session SMTP, la première commande à envoyer est la commande *HELO* suivie d'un espace (*SP*) et du nom de domaine de votre machine, puis valider par entrée (noté <*CRLF*>). Depuis avril 2001, les spécifications du protocole SMTP, définies dans le [RFC2821], imposent que la commande HELO soit remplacée par la commande *EHLO* ;
- La seconde commande est "*MAIL FROM:*" suivie de l'adresse email de l'expéditeur. Si la commande est acceptée le serveur renvoie le message "*250 OK*" ;
- La commande suivante est "*RCPT TO :*" suivie de l'adresse email du destinataire. Si la commande est acceptée le serveur renvoie le message "*250 OK*" ;
- La commande *DATA* est la troisième étape de l'envoi. Elle annonce le début du corps du message. Si la commande est acceptée le serveur renvoie un message intermédiaire numéroté *354* indiquant que l'envoi du corps du mail peut commencer et considère l'ensemble des lignes suivantes jusqu'à la fin du message repéré par une ligne contenant uniquement un point. Le corps du mail contient éventuellement certains des en-têtes suivants : Date, Subject, Cc, Bcc, From.

Si la commande est acceptée le serveur renvoie le message "*250 OK*"
Les spécifications de base du protocole SMTP veulent que tous les caractères transmis soient codés en code ASCII sur 7 bits et que le 8ème bit soit explicitement mis à zéro. Ainsi pour envoyer des caractères accentués il faut faire recours à des algorithmes d'encryptage des spécifications MIME :

- **base64** pour les fichiers attachés
- **quoted-printable** (d'abbréviation *QP*) pour les caractères spéciaux contenus dans le corps du message

L'ensemble des spécifications du protocole SMTP est défini dans le [RFC821].

A) Vulnérabilité de courrier électronique

Le courrier électronique est le service le plus répandu sur un réseau, du fait de son utilisation courante par la quasi-totalité des utilisateurs d'un réseau. C'est également l'un des services les plus vulnérable. Le serveur de courrier se compose de trois parties : un serveur, qui reçoit le courrier ou l'envoie, un agent de livraison, qui met le courrier dans la boîte de son destinataire, et un agent utilisateur, qui permet à l'utilisateur de lire son courrier et d'en écrire.

Trois types d'attaques sont susceptibles d'agir contre un serveur de courrier :

- **l'attaque par canal de commandes :** le serveur est vulnérable aux attaques par les commandes qu'il reçoit de l'extérieur ;
- **l'attaque par biais de données :** l'agent utilisateur et l'agent de livraison sont sensibles aux messages eux-mêmes ;
- **le bogue de lignes de commandes** : un programme peut faire l'objet d'une manipulation frauduleuse par quelqu'un qui arriverait à le contrôler [SEC 02].

On trouve aussi d'autre type d'attaque :

- **Le Mail Bombonig** : consiste à envoyer un nombre faramineux d'emails (plusieurs milliers) à un ou des destinataires.
- **L'attaque SMTPd overflow** : consiste à envoyer la commande « help » avec un argument trop long vers un serveur SMTP. Si le gestionnaire SMTP n'est pas patché pour prévenir de cette attaque, il plante.

3.4.3 PROTOCOLE SNMP

Le protocole SNMP est le langage que les agents et les stations de gestion (managers) utilisent pour communiquer. C'est un protocole de type question/réponse asynchrone. Ce protocole est situé au niveau application du modèle OSI, c'est lui qui définit la structure formelle des communications. SNMP est encapsulé dans des trames UDP. La MIB (Management Information Base) regroupe l'ensemble des variables relatives aux matériels et aux logiciels supportés par le réseau, et définit les objets de gestion dans l'environnement TCP/IP. La SMI (Structure of Management Information), définit comment sont représentées, dans la MIB, les informations relatives aux objets de gestion et comment sont obtenues ces informations.

SNMP a l'avantage d'être simple, cependant il a des capacités très limitées au niveau sécurité, principalement pour l'authentification. Tous les systèmes SNMP doivent également supporter les protocoles DUPER et IP pour transporter les données entre les agents et les stations de gestion.

A) Spécifications

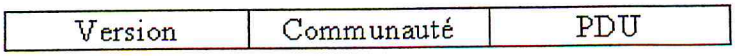


Figure 2. 16: Le format de la trame SNMP [TIL 99]

- **Version** : numéro de version SNMP. Le manager et l'agent doivent utiliser le même numéro ;
- **Communauté** : ce champ sert à identifier auprès du manager l'agent avant de lui accorder un accès ;
- **PDU** : il y a 5 types de PDU : GetRequest, GetNextRequest, GetResponse, SetRequest, et TRAP.

Un premier format est utilisé pour les PDU du genre GET, ou SET :

Type de PDU	ID de requête	Statut d'erreur	Index d'erreur	Obj 1, val 1

Figure 2. 17: Une description de ces PDU [TIL 99]

-Type de PDU :
0 : GetRequest,
1 : GetNextRequest,
2 : GetResponse,
3 : SetRequest ;
-ID de requête : champ qui coordonne la requête du manager et la réponse de l'agent ;
-Statut d'erreur : entier qui indique une opération normale « 0 » ou bien une erreur ;
-Index d'erreur : identifie les entées avec la liste des variables qui ont causé l'erreur ;
-Obj/Val : association du nom de la variable à transmettre avec sa valeur.

Type de PDU	Entreprise	Adresse Agent	Type Generique	Type Specifique	Timestamp	Obj 1, val 1

Figure 2. 18: Le second format utilisé pour la TRAP PDU [TIL 99]

-Type de PDU : dans ce cas toujours égal à 4,
-Entreprise : identifie l'entreprise de management qui a défini la Trap,
-Adresse Agent : adresse IP de l'agent,
-Type Générique : décrit quel type de problème est survenu (7 valeurs sont possibles),
-Type Spécifique : est utilisé afin d'identifier une TRAP spécifique à une entreprise,
-Timestamp : contient la valeur de l'objet sysUptime représentant le temps écoulé depuis la dernière initialisation,
-Obj/Val : association du nom de la variable à transmettre avec sa valeur.

B) SNMPv2 par rapport à SNMP

- **SNMPv2** est capable de gérer de manière distribuée un réseau : opérations entre stations d'administration,
- sécurité renforcée,
- nouvelles opérations.

La coexistence des 2 versions est facilitée par le fait que SNMPv2 est un sur ensemble de SNMPv1.

La manière la plus simple de gérer le passage de V1 à V2 est de passer la station d'administration à la version 2, qui peut ainsi gérer à la fois des stations en V2 (en cas de gestion répartie) et des agents en V1 et V2.

Il est nécessaire des équivalences dans : la manière dont sont gérées les informations (SMI) et le protocole.

C) La sécurité dans SNMP 2

Dans la version 1 => utilisation de la notion de communauté pour définir la visibilité accordée à une station par un agent.
Dans la version 2 => notion de groupe :

```
SnmpParty ::= SEQUENCE
{
partyIdentify OBJECT IDENTIFIER, -- identifiant du groupe
partyDomain OBJECT IDENTIFIER, -- type de couche transport
partyAddress OCTET STRING, -- adresse de niveau transport
partyMaxMessageSize INTEGER, -- taille max des messages
partyAuthProtocol OBJECT IDENTIFIER, -- nomme le protocole d'authentification utilisé
partyAuthClock INTEGER, -- période valide pour le groupe
partyAuthPrivate OCTET STRING, -- clé privée d'authentification
partyAuthPublic OCTET STRING, -- clé publique d'authentification
partyAuthLifeTime INTEGER, -- durée de vie des messages
partyPrivProtocol OBJECT IDENTIFIER, --identification du protocole utilisé
partyPrivPrivate OCTET STRING, -- clé privée
partyPrivPublic OCTET STRING, -- clé publique
}
```

Un élément actif sur le réseau agit de la manière suivante :

- exécute uniquement les opérations permises par le groupe,
- maintient une petite base de données qui contient tous les groupes reconnus par l'entité, les opérations pouvant s'effectuer directement et celles qui font appel à un agent de proxy, les ressources accessibles (notion de contexte).

Chaque entité maintient donc l'ensemble des données définissant le concept de "politique d'accès".

D) Format des messages sécurisés

privDest* authInfo *dstParty *srcParty *contexte* PDU = **Format général**
privDest* octet string* dstParty *srcParty *contexte* PDU = **Message non sécurisé**
privDest *digest *dst timestamp *src timestamp* dstParty *srcParty *contexte *PDU = **Authentifié non privé**
privDest *octet string *dstParty *srcParty *contexte *PDU*crypté = **Privé non authentifié**
privDest *digest *dst timestamp *src timestamp *dstParty *srcParty *contexte *PDU*crypté = **Privé et authentifié**

privDest : désigne le groupe pour lequel le message est destiné,
authInfo : protocole d'authentification utilisé.

3.4.4 PROTOCOLE FTP

Le protocole FTP (File Transfer Protocol) est, comme son nom l'indique, un protocole de transfert de fichier. Le protocole FTP est actuellement défini par le [RFC 959] (*File Transfer Protocol (FTP) - Specifications*), définit la façon selon laquelle des données doivent être transférées sur un réseau TCP/IP.
Le protocole FTP a pour objectifs de :

- permettre un partage de fichiers entre machine distante,
- permettre une indépendance aux systèmes de fichiers des machines clientes et serveur,
- permettre de transférer des données de manière efficace.

Le protocole FTP s'inscrit dans un modèle client-serveur, c'est à dire qu'une machine envoie des ordres (le client) et que l'autre attend des requêtes pour effectuer des actions (le serveur).

Lors d'une connexion FTP, deux canaux de transmission sont ouverts :

- Un canal pour les commandes (canal de contrôle) ;
- Un canal pour les données.

Lors de la connexion d'un client FTP à un serveur FTP, le USER-PI initie la connexion au serveur selon le protocole Telnet. Le client envoie des commandes FTP au serveur, ce dernier les interprètes, pilote son DTP, puis renvoie une réponse standard. Lorsque la connexion est établie, le serveur-PI donne le port sur lequel les données seront envoyées au Client DTP. Le client DTP écoute alors sur le port spécifié les données en provenance du serveur. Il est important de remarquer que, les ports de contrôle et de données étant des canaux séparés, il est possible d'envoyer les commandes à partir d'une machine et de recevoir les données sur une autre. Ainsi, il est par exemple possible de transférer des données entre deux serveurs FTP en passant par un client pour envoyer les instructions de contrôle et en transférant les informations entre deux processus serveurs connectés sur le bon port.

Attaque sur FTP

FTP Bounce : c'est un cas de spoofing d'adresse IP. Elle est basée sur une utilisation de la commande PORT du protocole FTP lorsque le serveur FTP est en mode actif. En effet, cette commande permet de se connecter à n'import quel autre serveur distant, et à un port donné. Dans ce cas, il est possible que la sécurité du serveur cible soit compromise, dans le cas ou il effectue une vérification des adresses IP d'origine. En effet, l'adresse IP que le serveur cible verra sera l'adresse IP du serveur FTP, et non de l'adresse IP de l'attaquant.

3.4.5 PROTOCOLE TELNET

Le protocole Telnet est un protocole standard d'Internet permettant l'interfaçage de terminaux et d'applications à travers Internet. Ce protocole fournit les règles de base pour permettre de relier un client (système composé d'un affichage et d'un clavier) à un interpréteur de commande (côté serveur).

Le protocole Telnet s'appuie sur une connexion TCP pour envoyer des données au format ASCII codées sur 8 bits entre lesquelles s'intercalent des séquences de contrôle Telnet. Il fournit ainsi un système orienté communication, bi-directionnel (half-duplex), codé sur 8 bits facile à mettre en oeuvre.

Le protocole Telnet repose sur trois concepts fondamentaux :

- Le paradigme du terminal réseau virtuel (NVT),
- Le principe d'options négociées,
- Les règles de négociation.

Ce protocole est un protocole de base, sur lequel s'appuient certains autres protocoles de la suite TCP/IP (FTP, SMTP, POP3). Les spécifications de Telnet ne mentionnent pas d'authentification car Telnet est totalement séparé des applications qui l'utilisent (le protocole FTP définit une séquence d'authentification au-dessus de Telnet). En outre le protocole Telnet est un protocole de transfert de données non sûr, c'est à dire que les données qu'il véhicule circulent en clair sur le réseau (de manière non chiffrée). Lorsque le protocole Telnet est utilisé pour connecter un hôte distant à la machine sur lequel il est implémenté en tant que serveur, ce protocole est assigné au port 23.

A) Vulnérabilité de service telnet

Généralement, les sites veulent autoriser leurs utilisateurs à employer des services sortants, pour qu'ils puissent accéder à des machines qui soient à l'extérieur du réseau local. En échange, on préfère contrôler les transactions Telnet entrant (de l'extérieur vers l'intérieur), pour les problèmes que posent ce genre de connexions :

- **le détournement :** quelqu'un pirate une connexion après que l'utilisateur se soit identifié auprès du système ;
- **l'espionnage de paquets :** quelqu'un lit les données qui transitent sans pour autant interférer avec le réseau ;
- **la fausse authentification :** quelqu'un essaye de se faire passer pour un utilisateur valide.

Dans le cas des sessions Telnet, entrant ou sortant, celles-ci sont sensibles à L'espionnage, car Telnet est un protocole en mode texte. Comme aucun échange n'est crypté, un éventuel espion peut récupérer des données confidentielles, ou bien encore des mots de passes [SEC 02].

B) La solution SSH

Il faut tout d'abord restreindre au maximum le Telnet entrant. Néanmoins, lorsqu'il est nécessaire d'avoir un service entrant, il faut mettre en place une politique d'identification la plus stricte possible. L'utilisation de Telnet comme service sortant pose peu de problèmes, surtout si on l'utilise à travers un pare-feu mandataire ou par filtrage de paquets. Si les données auxquelles on accède sont privées ou sensibles, on recommande l'utilisation d'un système d'accès à distance qui soit chiffré. C'est le cas de SSH. SSH (Secure SHell) procure un shell distant et sécurisé, qui va être utilisé à la place des commandes *telnet* ou encore des commandes 'r' (rsh, *rlogi*n…).

La commande SSH va être une solution adéquate pour des problèmes importants, tels que la circulation des mots de passe en clair sur le réseau, En utilisant SSH, on va

exécuter sans trop de crainte des commandes à distance, ou bien encore améliorer sensiblement la sécurité des transferts de données. SSH va utiliser un système de clés publiques pour renforcer l'authentification des sessions. SSH va véhiculer dans son canal de connexion n'importe quel flux reposant sur TCP. C'est à dire que SSH va encapsuler le flux qui ne passera plus par les ports auxquels il est accoutumé, mais par le tunnel créer par SSH. Il existe deux versions de SSH (SSH1 et SSH2) qui sont incompatibles, la version SSH2 étant mieux construite, et disposant de ce qui se fait de mieux en matière de cryptographique.

SSH utilise quatre clés cryptographiques. Les trois premières sont constituées d'une paire clé publique/clé privée [CHA 02].

- la clé utilisateur, qui permet l'authentification de l'utilisateur,
- la clé hôte, qui permet l'authentification des machines entre elles,
- la clé serveur, seulement utilisé par SSH1 pour sécuriser l'échange de la clé de session (SSH2 utilise un protocole particulier pour cela),
- la clé de session, qui est utilisée pour crypter le canal de communication.

3.4.6 LES SERVEURS DE NOM (DNS)

L'adresse IP numérique étant difficile à manipuler, une représentation hiérarchique de nom de machines a été mise en place pour faciliter l'utilisation du réseau. Cependant dans les couches basses du réseau seul la valeur numérique est utilisée. Le DNS est non pas une couche du réseau, mais une application. Les noms sont composés par une suite de caractères alphanumériques encadrés par des points. Par exemple www.univ-blida.dz correspond à l'adresse 192.50.125.2 et le mécanisme qui associe le nom au numéro s'appelle la résolution de noms. Cette représentation est hiérarchique.

Les domaines de la racine sont des domaines génériques ou des domaines géographiques.

- **DZ** : Algerie.
- **com :** Organisations commerciales,
- **edu** : Institutions éducatives,
- **gov :** Organisations gouvernementales,
- **int** : Organisations internationales,
- **mil** : Militaires,
- **net** : Réseau,
- **org** : Organisation à but non lucratif.

A) DNS spoofing

L'objectif de cette attaque est de rediriger des utilisateurs d'Internet vers des sites pirates, sans qu'ils ne s'en rendent compte. Pour cela, le pirate profite des faiblesses du protocole DNS (Domain Name System).

Il existe deux façons de faire une attaque DNS Spoofing

* **Le DNS ID Spoofing :** va consister à récupérer le numéro d'identification retourné dans la réponse d'une demande de résolution DNS, et d'envoyer une réponse à la machine qui a lancé la requête avant le serveur DNS auquel avait été envoyé la requête. Pour récupérer le numéro d'identification, on peut simplement écouter le réseau si on est dans le même réseau physique, ou bien en utilisant une faille des systèmes d'exploitation ou des serveurs DNS, par exemple si le serveur retourne un numéro qui soit prédictible [SEC 02].

* **Le DNS Cache Poisoning :** se place à un autre niveau. Les serveurs DNS possèdent un cache dans lequel ils gardent, pendant un certain temps, la correspondance entre un nom de machine et son adresse IP. Le serveur DNS ne connaît les correspondances que pour les machines de son réseau. Le cache lui permet donc de ne pas interroger sans cesse le serveur DNS du domaine à contacter, mais de regarder directement dans son cache, ce qui est beaucoup plus rapide.

Le DNS Cache Poisoning consiste à corrompre ce cache avec de fausses informations. Pour cela, le pirate doit avoir sous son contrôle un nom de domaine (par exemple *pirate.com*), et le serveur DNS ayant autorité sur celui-ci. L'attaque se déroule ainsi :

- le pirate envoie une requête vers le serveur DNS cible demandant la résolution du nom d'une machine de son domaine;
- le serveur DNS relaie cette requête au serveur DNS de *pirate;*
- le serveur DNS du pirate enverra, en plus de la réponse, des informations additionnelles, dans lesquelles sera précisé le nom de la machine publique avec l'adresse IP du pirate ;
- Les enregistrements additionnels sont ajoutés dans le cache du serveur DNS cible ;
- Une machine qui demandera alors au serveur DNS cible de résoudre un des noms corrompus aura l'adresse IP associée à la machine pirate, et pas la bonne concordance.

Dans tous les deux cas, le pirate répond avant le serveur DNS, pour cela il dirige un déni de service contre le serveur (attaque DoS).

4. LES FAILLE DE SECURITE DANS LES SYSTEMS

Après qu'on a vu l'architecture de la suite de protocole TCP/IP et leurs vulnérabilités, on va décrire d'autre type de vulnérabilité dans un réseau due au system à savoir les failles suite au mal configuration ou bien des bugs dans les system et ainsi négligence d'une politique de sécurité, on va baser sur les systèmes d'exploitations Windows.

4.1 FAILLES SUITE À LA CONFIGURATION DU SYSTEM

4.1.1 BRUTE FORCE CRACKING

Ce procédé consiste à tester de façon exhaustive toutes les combinaisons possibles de caractères (alphanumériques + symboles), de manière à trouver au moins un mot de passe valide. Le hacker peut faire des brutes forces attaques sur n'importe quel compte sans restrictions. Pour éviter cela on doit activer le "LockOut Account" qui bloque le système au bout d'un nombre limité de tentatives, Microsoft recommande d'activer le Lockout Account après cinq essais incorrects [SEC 02].

4.1.2 ATTAQUE +++ATHZERO

L'attaque +++ATH0 vise certains modems compatibles Hayes. Lorsque ce type de modem reçoit la commande +++ATH0, il risque de se déconnecter. En effet, cette commande permet de positionner le modem en commande manuelle. En pratique, cela se traduit par l'envoi d'un « ping » contenant la chaîne de caractères « +++ATHp ». Pour ce protéger de cette attaque on doit rechercher dans la base de registre la clé : HKEY_LOCAL_MACHINE\Système et créer la chaîne « UserInit », ayant pour valeur « s2=255 ».

4.1.3 REGISTRE NT

L'OS 32-bit de Windows enregistre la plupart des informations spécifique au système à l'intérieur de la base de registre de Windows .Les seules personnes qui ont normalement accès a la base de registre devrais être les Administrateurs et les Superusers. Techniquement les utilisateurs qui doivent installer des programmes ou qui doivent surveiller les audits ou les autres informations systèmes devraient avoir accès à la base de registre de Windows NT. Si des utilisateurs ont des droits d'accès, ou si l'utilisateur Guest a des privilèges, des utilisateurs non autorisés et des hackers peuvent éditer la base de registre de Windows. Un utilisateur avec le privilège d'éditer la base de registre de Windows NT peut fondamentalement changer le setup de l'installation du serveur Windows NT. Après qu'un intrus ouvre la base de registre de Windows NT, il peut facilement lire la liste des contrôles d'accès depuis les clés de registres : HKEY_LOCAL_MACHINE et HKEY_CLASSES_ROOT.

Si le hacker a des droits pour éditer a l'intérieur de la base de registre NT, il peut changer chaque entrée (programmes, fichiers associés ...) avec une permission en écriture pour lui, il peut altérer les fichiers associés tel que un .txt ne lance plus notepad.exe mais un Trajan ou un autre programme pour modifier ou faire rebooter le système. De plus si le compte Guest a un accès en écriture sur la base de registre soit Windows NT est très mal configuré soit un intrus est déjà passé par la. Il est conseiller de revoir les permissions pour la base de registre assez souvent .Sous Windows NT 4.0 on empêche l'accès a la base de registre par le réseau en ne laissant que les utilisateurs ayant physiquement accès a la console éditer la base de registre.

4.1.4 ACCOMPTE GUEST SANS PASSWORD

Bien que Windows NT 4.0 ai le compte "Guest" non activé par défaut,les versions 3.5 et 3.51 incluent un account "Guest" global. Si on désactive pas ce compte "Guest" et qu'il n'ai pas de mot de passe, un hacker peut se loguer dans le serveur avec n'importe quel login et password. Si on ne restreint pas l'accès à certains fichiers ainsi qu'aux registres NT, l'intrus peut alors accéder à des zones sensibles. En clair, la meilleure solution à ce problème est de désactiver le compte "Guest".

4.2 LE MODULE NETBIOS

(Network basic input/output system) de Microsoft Windows est utilisé par des programmes pour les fonctions de réseau local (gestion de noms, sessions, etc.). Une alerte de vulnérabilité dans l'un des services NetBIOS over TCP existe :
Le **NetBIOS Name Service** (NBNS) équivalent des DNS en TCP/IP. NBNS permet par exemple de trouver l'adresse IP d'un système à partir de son nom NetBIOS. Dans ce service, la réponse à une requête peut inclure des données aléatoires issues de la mémoire du système cible. Un pirate peut donc exploiter cette faille en envoyant une requête NBNS sur le port 1031 et en analysant la réponse obtenue pour obtenir des données issues de la mémoire du système de la victime. cette vulnérabilité concern Windows NT 4.0 Server, Windows NT 4.0, Terminal Server Edition, Windows 2000, Windows XP et Windows Server 2003 [MIC 01].
- **l'attaque Out OF Band** (OOB) : est plus connue sous le nom de « Nuke ». Elle vise le port 139 (netbios Session Service port). Lorsqu'un packet est envoyé sur le port 139 avec le flag « urgent », Windows attend des données qui doivent suivre le flag. S'il n'y a pas de données qui arrivent, le système ne sait pas gérer cette absence... ; pour arrêter cette attaque on peut bloquer le port 139 [SEC 02].

4.3 FAILLES SUITE AU BUG DU SYSTEM

4.3.1 LES TROUS DE SECURITE APPLICATIFS

Un trou de sécurité applicatif est le résultat d'un fonctionnement anormal d'une application. Il en résulte un plantage de l'application, ou bien un état non stable. Concrètement, il s'agit de trouver un fonctionnement que n'a pas prévu le programmeur. De ce fait, il est parfois posible d'en exploiter des failles. Cela devient très intéressant lorsque c'est un programme réseaux (architecture distribuée...) ;

4.3.2 LES BUFFERS OVERFLOW

Le fonctionnement général d'un buffer overflow est de faire crasher un programme en écrivant dans un buffer plus de données qu'il ne peut en contenir (un buffer est un zone mémoire temporaire –pile stack- utilisée par une applications), dans

le but d'écraser des parties du code de l'application et d'injecter des données utiles pour exploiter le crash de l'application.

4.3.3 Déni de service

Le « Denial-of-service » est une attaque très évoluée visant à rendre muette une machine en la submergeant de trafic inutile. Il peut y avoire plusieurs machines à l'origine de cette attaque (DDoS) qui vis à anéantir des serveurs, des sous-réseaux, etc. d'autre part, elle reste très difficile à conter ou à éviter. Parmi les attaques propres à créer un déni de service, nous pouvons rappeler les suivantes :

–***WINS 137/53 flood***, vise les systèmes *Windows* et consiste à envoyer un paquet *UDP* sur le port 137 ou des paquet TCP sur le port 53 ce qui arrête le service *WIN*S.

–***Snork***, vise les systèmes *Windows NT* et consiste à envoyer une trame *UDP* sur certains services qui, s'ils sont lancés, provoque une connexion d'une durée infinie ce qui limite la bande passante et la disponibilité du serveur.

-Attaque BrKill : elle consiste à générer des packets qui génèrent un reset, permettant à l'attaquant de couper la connexion de la victime, à distance. Les transferts dits connectés (FTP, telnet, ...) sont alors les cibles potentielles de cette attaque.

-attaque Coke : vise les systèmes qui exécutent le service WINS (Windows Internet Name Service). Elle consiste à se connecter à la cible et à envoyer n'importe quoi. En fonction de la configuration de l'ordinateur cible, celui-ci inscrira un message d'erreur dans le log pour chaque packet invalide reçu. Ceci a pour but de ralentir le système.

-NT Stop, vise les système *Windows NT 4*.0, consiste à envoyer une requête *SMB* 10 logon avec une taille spécifiée incorrecte qui génère une corruption de mémoire, provoquant le plantage de l'application.

– ***NT Inetinfo***, qui consiste à se connecter sur un serveur *Windows NT 4.0* sur le port 1031. Le processus inetinfo utilise alors beaucoup de ressources et peut provoquer un plantage ou un reboot [SEC 02].

5. CONCLUSION

Le modèle OSI reste un modèle de référence permettant de comprendre le rôle de chaque élément du réseau. Alors que TCP/IP est désormais reconnu unanimement comme le protocole de communication prédominant pour interconnecter différents systèmes informatiques. Cependant Les vulnérabilités sont, en fait, des points faibles qui sont entrain d'être fixés (dans certain cas, il le sont déjà). Mais certaines d'entre elles font partie de la philosophie de design de TCP/IP.

Ainsi les failles des system et surtout les déférentes applications réseaux installés ouvrent une grande porte aux intrusions. Pour cela en va étudier dans le chapitre suivant, les solutions existent pour ces problèmes de sécurité.

CHAPITRE III

SOLUTIONS DE SECURITE

1. INTRODUCTION

Il est clair que la sécurité est un des problèmes les plus sérieux que connaissent les entreprises qui ont des réseaux informatiques. La difficulté est que chaque problème de sécurité à une solution qui lui est propre, et qui ne fonctionnera que pour ce problème. De plus, on ne connaît pas tous les problèmes qui peuvent conduire à des failles de sécurité, la plupart du temps on ne connaît le problème et sa solution q'une fois le problème survenu. Cela est rendu plus compliqué encore par la découverte quotidienne de nouvelles méthodes pour exploiter ces failles. Suite à cela il est nécessaire de définir une politique de sécurité.

Comme nous l'avons veut dans le premier chapitre l'établissement d'une telle politique de sécurité est le premier pas vers un réseaux sécurisé, c'est aussi l'étape la plus dure à mettre en place et la plus importante.

1.1 MISE EN PLACE D'UNE POLITIQUE DE SECURITE

La sécurité informatique doit toutefois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps les règles de sécurité, c'est à dire :

- La politique de sécurité,
- Organisation de la sécurité,
- Classification et contrôle des actifs,
- Sécurité des ressources humaines,
- Sécurité physique et sécurité de l'environnement,
- Exploitation et réseaux,
- Contrôle d'accès,
- Développement et maintenance des systèmes,
- Continuité de service,
- Conformité.

Pour plus de détail voire l'annexe A « ISO 17799 ».

De nombreux outils existent pour améliorer la sécurité d'un réseau hétérogène. Beaucoup de ces outils sont commerciaux, mais l'émergence et l'engouement actuel pour les solutions dites libres ont facilité l'apparition des produits de bonne qualité. Il ne s'agit pas ici de faire une liste de tous ces outils mais d'en voir les caractéristiques générales des outils les plus connus.

Les outils mis en place par la suite respecteront cette politique de sécurité, et devront même la refléter, comme dans ce modèle classique.

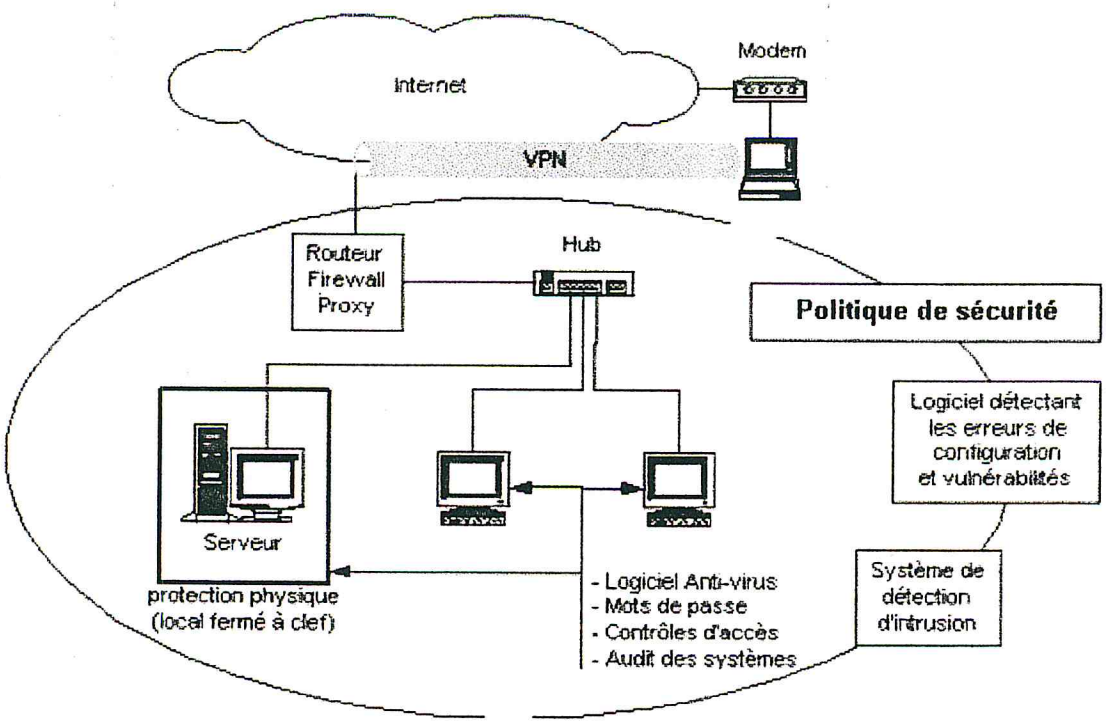


Figure 3. 1: outils de sécurité [GUI 00]

On va étudier les trois principaux outils qui permettent d'assure la sécurité des réseau les VPN, les FIREWALL et les IDS.

2. LES VPN (VIRTUAL PRIVATE NETWORK)

Les réseaux privés virtuels permettent à l'utilisateur de créer un chemin virtuel sécurisé entre une source et une destination. Avec le développement d'Internet, il est intéressant de permettre ce processus de transfert de données sécurisé et fiable. Grâce à un principe de tunnel (tunnelling) dont chaque extrémité est identifiée, les données transitent après avoir été chiffrées. Un des grands intérêts des VPN est de réaliser des réseaux privés à moindre coût. En chiffrant les données, tout se passe exactement comme si la connexion se faisait en dehors d'Internet. Les données à transmettre peuvent appartenir à un protocole différent d'IP. Dans ce cas le protocole de tunnelling encapsule les données en rajoutant une entête. Permettant le routage des trames dans le tunnel. Le tunnelling est l'ensemble des processus d'encapsulation, de transmission et de désencapsulation. La sécurité des échanges est assurée à plusieurs niveaux et par différentes fonctions comme le cryptage des données, l'authentification des deux extrémités communicantes et le contrôle d'accès des utilisateurs aux ressources.
Il existe sur le marché quatre principaux protocoles [GUI 00] :

- PPTP (Point to Point Tunnelling Protocol) de Microsoft,
- L2F (Layer Two Forwarding) de Cisco,
- L2TP (Layer Two Tunnelling Protocol) de l'IETF,
- IPSec (version protégée du protocole IP).

2.1 PPTP (POINT TO POINT TUNNELLING PROTOCOL)

C'est un protocole de niveau 2 qui encapsule des trames PPP dans des datagrammes IP afin de les transférer sur un réseau IP. PPTP permet le cryptage des données PPP encapsulées mais aussi leur compression.
Le schéma suivant montre comment un paquet PPTP est assemblé avant d'être transmis par un client distant vers un réseau cible.

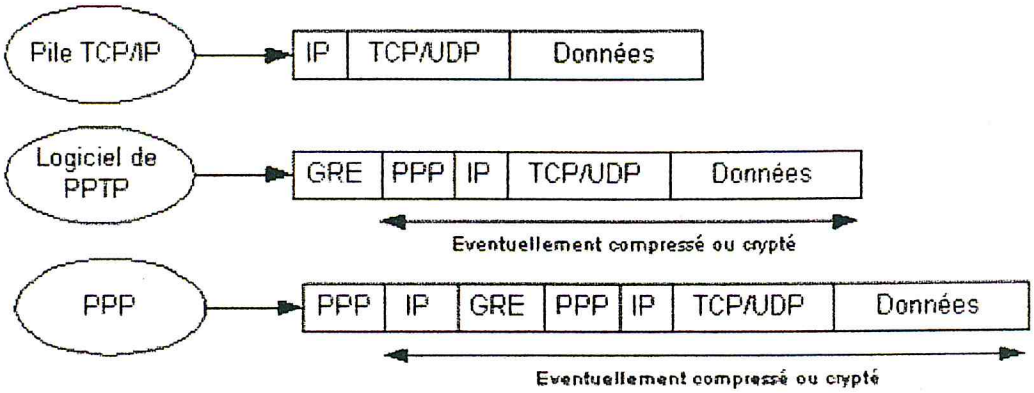


Figure 3. 2: L'assemblage d'un paquet PPTP [GUI 00]

L'intérêt de PPTP est de ne nécessiter aucun matériel supplémentaire car les deux logiciels d'extrémité (le client et le serveur) sont intégrés dans NT4. Par contre, il ne fonctionne que sous NT pour le moment.

2.2 L2F (LAYER TWO FORWARDING) :

L2F est un protocole de niveau 2 qui permet à un serveur d'accès distant de véhiculer le trafic sur PPP et transférer ces données jusqu'à un serveur L2F (routeur). Ce serveur L2F désencapsule les paquets et les envoie sur le réseau. Il faut noter que contrairement à PPTP et L2PT, L2F n'a pas besoin de client.
Ce protocole est progressivement remplacé par L2TP qui est plus souple.

2.3 L2TP (LAYER TWO TUNNELLING PROTOCOL)

Microsoft et Cisco, reconnaissant les mérites des deux protocoles L2F et PPTP, se sont associés pour créer le protocoles L2TP. Ce protocole réunit les avantages de PPTP et L2F. L2TP est un protocole réseau qui encapsule des trames PPP pour les envoyer sur des réseaux IP, X25, relais de trames ou ATM. Lorsqu'il est configuré pour transporter les données sur IP, L2TP peut être utilisé pour faire du tunnelling sur Internet. Mais L2TP peut aussi être directement mis en œuvre sur des supports WAN (relais de trames) sans utiliser la couche de transport IP.
On utilise souvent ce protocole pour créer des VPN sur Internet. Dans ce cas, L2TP transporte des trames PPP dans des paquets IP. Il se sert d'une série de messages L2TP pour assurer la maintenance du tunnel et d'UDP pour envoyer les trames PPP dans du L2TP [GUI 00].

2.4 PROTOCOLE IPSEC

IPSEC (IP Sécurisé) interdit la translation d'adresse au milieu d'un tunnel VPN, Afin de se prémunir des écoutes, et des usurpations d'identité, le groupe IPSec de l'IETF a modifié le protocole IPv4 pour permettre de [LAU 01] :

- chiffrer les paquets IP (leur contenu seul ou le paquet entier) ;
- introduire un authentificateur permettant d'authentifier l'équipement émetteur et de contrôler l'intégrité de tout ou parties du paquet IP.

Ainsi, deux nouveaux en-têtes IPv4 dédiés à la sécurité ont été définis (ils sont encore appelés extensions pour la nouvelle génération IPv6 du protocole IP) :

- **l'en-tête d'authentification** (AH : Authentication Header) : contient entre autre un authentificateur. Il permet donc de vérifier l'identité de l'équipement de sécurité émetteur, et de contrôler l'intégrité des données. Il permet optionnellement de détecter si les données reçues n'ont pas été précédemment reçues (afin d'éviter qu'une personne malveillante ayant réussi à récupérer un paquet légitime sur le réseau ne puisse rejouer un paquet vers le même destinataire) ;
- **l'en-tête de confidentialité** (ESP : Encapsulating Security Payload) : permet de transporter un paquet IP tout ou en partie chiffré, ainsi qu'un authentificateur offrant la même gamme de services que l'en-tête d'authentification [RFC 2402].

2.5 INCONVENIENT DE VPN

LE VPN n'est pas encore une technologie assez mature et les solutions proposées sur le marché à l'heure actuelle ne permettent que des garanties sur des réseaux locaux propriétaires.

3. LES PARE-FEU (FIREWALL)

Un firewall, est un système physique ou logique servant d'interface entre un ou plusieurs réseaux afin de contrôler et éventuellement bloquer la circulation des paquets de données, en analysant les informations contenues dans les couches 3, 4 et 7 du modèle OSI. Il s'agit donc d'une machine (machine spécifique dans le cas d'un firewall matériel ou d'un ordinateur sécurisé hébergeant une application particulière de pare-feu) comportant au minimum deux interfaces réseau :

- une interface pour le réseau à protéger (réseau interne)
- une interface pour le réseau externe

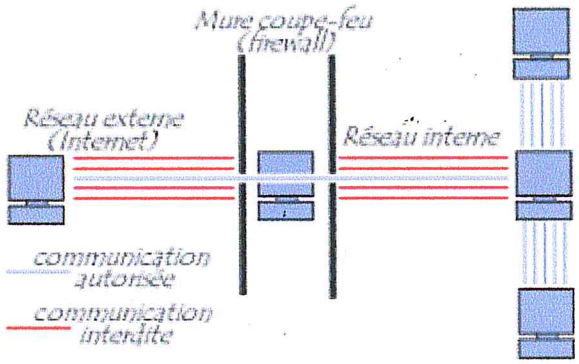


Figure 3. 3: un firewall [PIL 03]

Le pare-feu représente ainsi généralement dans les entreprises un dispositif à l'entrée du réseau qui permet de protéger le réseau interne d'éventuelles intrusions en provenance des réseaux externes (souvent Internet). Les terminologies suivantes sont parfois également utilisées : garde-barrière (gate-keeper), porte coupe-feu (firewall), antéserveur, écluse... etc.

Lorsque certaines machines du réseau interne ont besoin d'être accessible de l'extérieur (comme c'est le cas par exemple pour un serveur Web, un serveur de messagerie, un serveur FTP public, ...) il est souvent nécessaire de créer une nouvelle interface vers un réseau à part, accessible aussi bien du réseau interne que de l'extérieur, sans pour autant risquer de compromettre la sécurité de l'entreprise. On parle ainsi de **zone démilitarisé sécurisée** (souvent notée **DMZ** pour *DeMilitarized Zone*) pour désigner cette zone isolée hébergeant des applications mises à disposition du public.

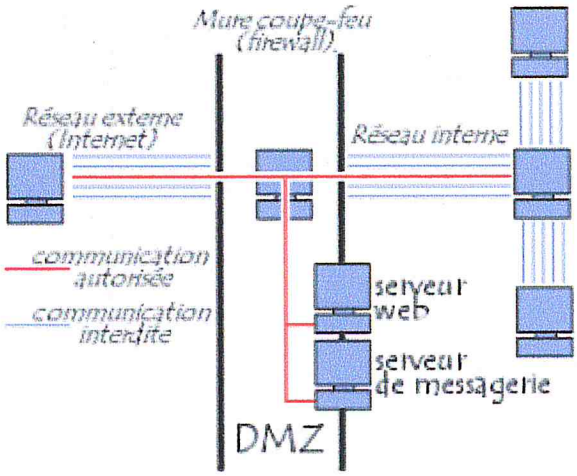


Figure 3. 4: firewall avec une DMZ [PIL 03]

Dans le cas où la zone protégée se limite à l'ordinateur sur lequel le firewall est installé on parle de firewall personnel [PIL 03].

3.1 LE FONCTIONNEMENT D'UN SYSTEME FIREWALL

Un système pare-feu contient un ensemble de règles prédéfinies permettant :

- Soit d'autoriser uniquement les communications ayant été explicitement autorisées *"Tout ce qui n'est pas explicitement autorisé est interdit"*. C'est un filtre avec machine bastion (par défaut) :
  - tout ce qui rentre est par défaut interdit, il faut donc autoriser explicitement,
  - tout ce qui sort du réseau interne vers le réseau externe est permit;
- Soit d'empêcher les échanges qui ont été explicitement interdits *"Tout ce qui n'est pas explicitement interdit est autorisé"* C'est un filtre minimum.

Le choix de l'une ou l'autre de ces méthodes dépend de la politique de sécurité adoptée par l'entité désirant mettre en oeuvre un filtrage des communications. La première méthode est sans doute la plus sûre, mais elle impose toutefois une définition précise et contraignante des besoins en terme de communication [GRE 03].

3.2 LE FILTRAGE DE PAQUETS :

Un filtre de paquets (*packet filter*) est un système multiport placé en coupure entre deux réseaux. Il analyse chaque paquet IP entrant et décide s'il doit être transmis sur le réseau ou détruit. Le filtrage s'effectue en fonction des informations habituellement trouvées dans l'en-tête des paquets IP :

- les numéros de protocole ;
- les adresses IP source et/ou destination ;
- les numéros de ports source et/ou destination (qui sont placés dans le message TCP ou UDP) ;
- les drapeaux de connexion TCP ;
- d'autres options.

3.2.1 FILTRAGE STATIQUE OU STATELESS INSPECTION

Le filtrage statique est une des premières solutions firewall à avoir été mise en oeuvre. Cette solution permet de déterminer la nature du service demandé et de définir si le paquet IP doit être accepté ou rejeté en fonction des règles définies. Le principal intérêt du filtrage statique réside dans sa transparence vis-à-vis des postes utilisateurs, ainsi que dans la vitesse des traitements.

Par exemple, une première règle indique que toutes les machines peuvent se connecter à un serveur Web sur Internet sur le port 80, et la suivante autorise le serveur Web à répondre à tous les clients du service (sur un port supérieur à 1024). Ces règles permettent à toutes les machines du réseau local d'accéder au Web.

Il est généralement impossible de gérer de façon satisfaisante les déférentes types de protocoles sans ouvrir l'accès à un plus grand nombre de ports, et donc de rendre le réseau plus vulnérable. Le filtrage dynamique répond à ces limites [GRE 03].

3.2.2 FILTRAGE DYNAMIQUE OU STATEFUL INSPECTION

Le filtrage dynamique reprend le principe de travail du filtrage statique au niveau de la couche réseau, ainsi que la transparence de sa mise en place. Or, de nombreux services (le FTP par exemple) initient une connexion sur un port statique, mais ouvrent dynamiquement un port de manière aléatoire afin d'établir une session entre la machine faisant office de serveur et la machine cliente. Ainsi, il est impossible de prévoir les ports à laisser passer ou à interdire. Pour y remédier, l'entreprise CHECK POINT SOFTWARE TECHNOLOGIES a breveté un système de filtrage dynamique de paquets ou stateful inspection basé sur l'inspection des couches réseau et transport du modèle OSI. Cette technologie permet d'effectuer un suivi des transactions entre le client et le serveur et donc d'assurer la bonne circulation des données de la session en cours. Le filtrage dynamique tient donc à jour une table des connexions ouvertes par les clients pour certaines applications. C'est pour cette table gérée de manière dynamique qu'il porte son nom [GRE 03].

Si le filtrage dynamique est plus performant que le filtrage statique de paquets, il ne protège pas pour autant de failles applicatives, c'est à dire les failles liées aux logiciels, représentant la part la plus importante des risques en terme de sécurité.

3.3 PASSERELLES APPLICATIVES

Une passerelle applicative (*application gateway*) est un serveur permettant d'effectuer un contrôle d'accès plus ou moins fin sur les données échangées entre deux réseaux pour un service TCP/IP particulier. Plus précisément, dans le modèle client-serveur, une passerelle est un serveur placé entre le client qui demande un service particulier et le serveur rendant ce service. Dans ce modèle, la passerelle fonctionne comme un serveur du point de vue du client et comme un client du point de vue du serveur. Au lieu d'avoir une seule connexion entre le client et le serveur, le serveur intercepte la connexion, effectue son filtrage, et si la connexion est autorisée, il relaie le trafic sur le serveur de destination tout en continuant à le filtrer si nécessaire. Ainsi, au lieu d'une connexion TCP/IP, il y en a deux : une entre le client et la passerelle et une autre entre la passerelle et le serveur.

Deux types de passerelles applicatives existent :

- les **passerelles de niveau applicatif,** encore appelées *Proxy*, permettent de faire un filtrage fin en fonction du service demandé : Telnet, FTP, SMTP ou HTTP. Du filtrage sémantique est ainsi rendu possible, ce qui permet par exemple de faire de la détection de fuites d'informations. Cependant, ces passerelles nécessitent que pour chaque service filtré, un serveur soit développé, ce qui est très lourd et peut être un frein au développement de nouveaux services. De plus, elles ne sont pas transparentes pour les utilisateurs en général puisque souvent, il faut qu'ils

s'authentifient auprès de la passerelle et qu'ils modifient la configuration de leur navigateur de telle sorte que le trafic associé à une application particulière soit orienté vers le proxy approprié ;

- les **passerelles de niveau circuit** filtrent au niveau de la couche transport. Elles offrent l'avantage d'être communes à toute application TCP/IP, contrairement aux passerelles de niveau applicatif. Elles sont totalement transparentes pour les utilisateurs, si ce n'est que la passerelle introduite un temps d'ouverture de connexion plus important. Par contre, le filtrage ne porte que sur le niveau transport. Pour les passerelles de niveau circuit, le contrôle effectué par le serveur peut prendre plusieurs formes, comme :
  - autoriser une connexion sur un port pour une durée maximale fixée ;
  - n'autoriser la réutilisation d'un même port qu'après un certain délai ;
  - authentifier un terminal, etc.

Pour cela, la politique de sécurité appliquée dans le pare-feu doit définir tous les paramètres adéquats comme les délais, les durées maximales de connexion, la liste des terminaux ayant l'autorisation de se connecter sur un port particulier, les caractéristiques permettant d'authentifier un terminal, etc [LAU 01].

3.4 LES LIMITES DES FIREWALLS :

Le fait d'installer un firewall n'est bien évidemment pas signe de sécurité absolue. Les firewall ne protège en effet que des communications passant à travers eux. Ainsi, les accès au réseau extérieur non réalisés au travers du firewall sont autant de failles de sécurité. Un firewall ne protège pas non plus des attaques au sein du réseau local. Si un pirate à l'intérieur du réseau local veut attaquer une machine au sein du même réseau local, le firewall (étant donné que les messages n'y transitent pas) ne sera d'aucune utilité. Les pare-feu ne permettent pas de filtrer les applications pour lesquelles il n'existe pas de passerelles applicatives. De plus, la protection offerte par les pare-feu est très limitée vis-à-vis des attaques portant sur le contenu du trafic, comme des programmes, des applets java ou des fichiers infectés par un virus de type « sendmail » comme « I love you ». Le pare-feu se contente de suivre les règles de sécurité qui lui auront été édictées. Il ne peut donc pas intervenir sur des menaces d'un type nouveau qui n'entreraient pas dans le cadre des règles. Il va donc être important de connaître les évolutions des attaques et de faire évoluer le pare-feu en conséquence.

Enfin, il fut bien voir que le pare-feu ne protège pas de certains types d'intrusions. En effet, ceux-ci vont transiter par des paquets de données, mais le pare-feu n'est pas apte à étudier les paquets qui transitent et à déduire ce qu'ils peuvent faire une fois que tous les paquets d'un même programme se seront réassemblés. Là encore, ce n'est pas le rôle du pare-feu d'agir à ce niveau, mais d'un system de détection d'intrusion.

4. LES IDS

La détection d'intrusion est une technologie de sécurité complémentaire des autres mécanismes mis en œuvre dans le cadre sécurité globale (authentification, chiffrement, outils de test de vulnérabilités, Firewall). Elle a pour objectif de détecter et d'isoler les «intrusions» contre les systèmes informatiques en contrôlant dynamiquement les actions des utilisateurs (externes et internes) du réseau. La mise en œuvre de cette fonctionnalité est réalisée par les outils de détection d'intrusion (Intrusion Détection Système).

Pour de nombreuses organisations qui viennent de déployer une technologie à barrière de protection au premier lieu de leur réseau, les systèmes de détection d'intrusion (IDS) constitue la prochaine étape logique.

Les IDS sont des systèmes qui collectent des informations de différentes manières, soit à partir du système ou du réseau, sur les différentes activités se rapportant à ce même système pour les analyser à la recherche de signes d'une intrusion (attaques suspectes) et alerter dans le cas positif [BAC 00].

Un IDS est en mesure d'assurer la protection requise contre les attaques internes (le trafic ne traverse absolument pas la barrière de protection) mais aussi il peut sécuriser le réseau contre les attaques externes.

IL existe plusieurs modèles d'IDS, et presque chacun a sa propre architecture. Donc Il est utile d'avoir un modèle (de référence) qui représente ces différents IDS.

4.1 LE MODEL DE REFERENCE CIDF

Le CIDF (*Common Intrusion Detection Framework*) est un groupe de travail sur la détection d'intrusion créé par DARPA (*Defense Advanced Research Projects Agency*) ; le CIDF défini un ensemble de composants qui, ensemble, définissent un IDS. Ces composants sont :

- Le Générateur d'événements *events generators* (E-boxes),
- L'analyseur d'événements *analysis engines* (A- boxes),
- Le mécanisme de stockage storage mechanism(D-boxes),
- Le gestionnaire des mesures conservatoires *events contre mesures*(C–boxes).

Comme la montre la figure 3.5

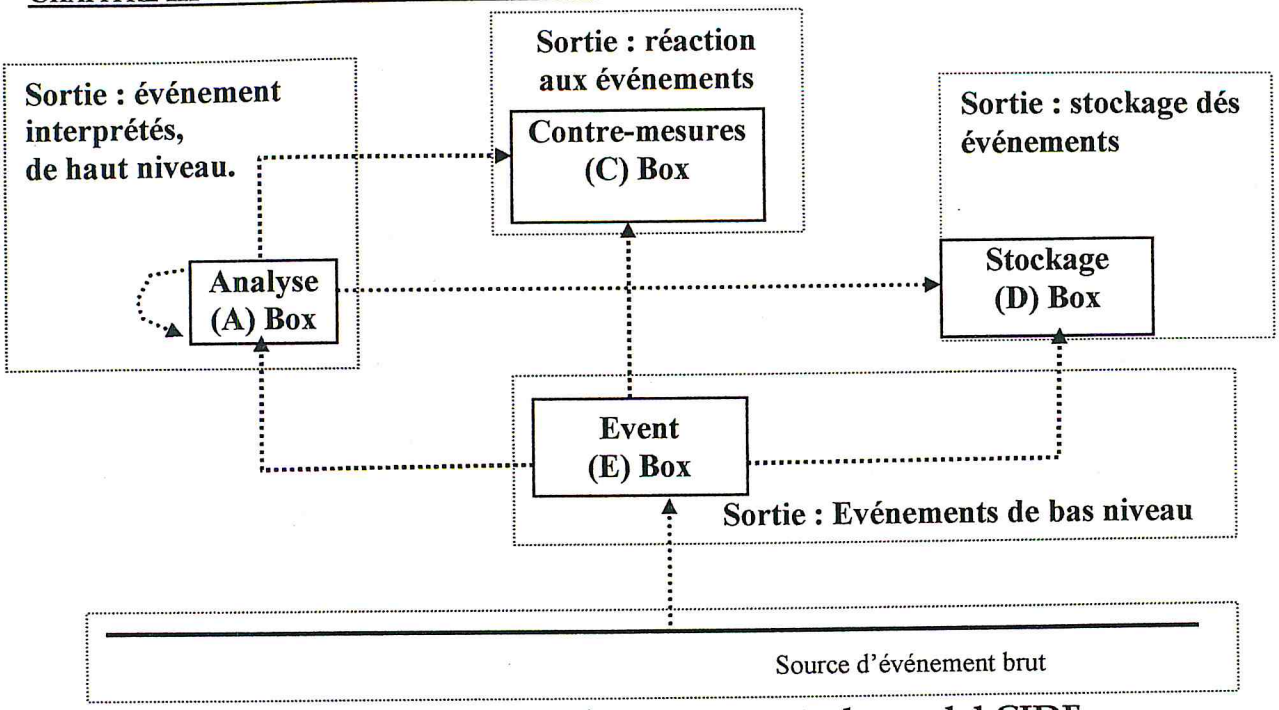


Figure 3. 5: Relations entre les composants du model CIDF.

Le rôle des déférents composants du model CIDF est :

- **Le générateur d'événement (E-BOX)** : Le E-box fournit des informations brutes sur l'état du système, du réseau, etc. C'est l'organe sensoriel de l'IDS. Sans la E-box un IDS n'a pas d'informations sur ce qui se passe (tant au niveau du réseau qu'au niveau de son propre état);
- **L'analyseur d'événements (A-BOX)** : Le A-box permet le réassemblage des informations reçues de l'E-box et assure l'analyse et l'identification de comportements à risque. Différentes approches sont étudiées pour l'analyse du flux d'événements, comme nous le verrons par la suite, plus un système est complexe et plus il a de machines à surveiller, plus les algorithmes de l'A-box doivent être performants. C'est donc un thème de recherche décisif (algorithme génétique, anomalie statistique, etc.). La connaissance du fonctionnement de l'A-box constitue bien sûr, pour les pirates, une information déterminante pour l'élaboration de stratégies visant à contourner, à leurrer ou à planter l'IDS;
- **Gestionnaire de mesures conservatoires (C-BOX)** : Plusieurs IDS sont conçus pour être un outil d'alarme. Cependant, la plupart des IDS commerciaux ajoute la fonctionnalité de contre-mesure. Le C-box est le bras séculier de l'A-box ; elle se charge de couper les connexions, de restreindre les accès jugés dangereux. Cette fonction permet aux IDS de se prémunir en temps réel contre des attaques depuis un site déjà identifié ou suspect selon des critères définis par l'administrateur réseau du site;
- **Le mécanisme de stockage (D-BOX)** : Le D-box assure la journalisation des événements. Cette fonction permet d'avoir les informations à disposition pour le *reporting* et de conserver une trace disponible à long terme.

Le document du CIDF n'a pas pour but de spécifier la méthode de développement de l'IDS mais plutôt de définir les interfaces qui doivent être supportées et leur organisation au sein de ces quatre fonctions. En effet, nous pourrions très bien imaginer un IDS complexe fonctionnant sur plusieurs machines : une ou plusieurs E-box (complémentaires ou redondantes), une ou plusieurs D-box (complémentaires ou redondantes) et un ensemble A-box C-box, le tout relié par des canaux spécifiques (clusters, etc.). Le modèle CIDF reste un modèle de référence. Les IDS de terrain sont surtout définis par des composants fonctionnels.

4.2 LES COMPOSANTS D'UN IDS

Les fonctionnalités d'un IDS peuvent être logiquement distribuées en trois composants logiques : les sondes, les analyseurs, l'interface utilisateur. Ces composants sont communs à tous les modèles qui existent [ALL 00].

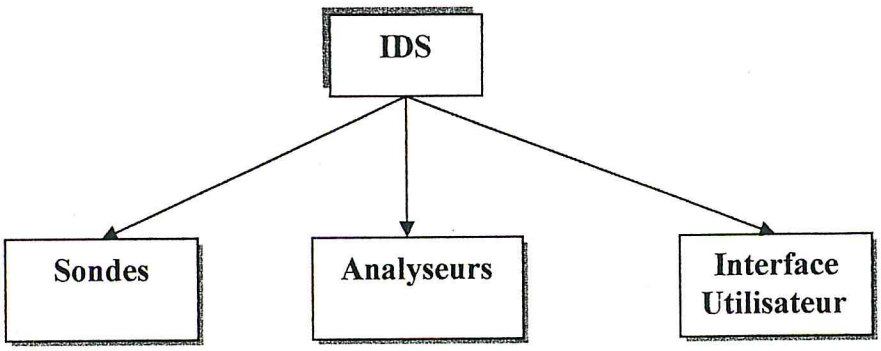


Figure 3. 6: les composants d'un IDS

- **Les sondes (Sensors)** : Ce module s'occupe de la collecte d'informations. L'entrée de la sonde peut être n'importe quelle partie du système qui peut contenir des informations pouvant nous permettre de détecter une intrusion. Les exemples de types d'entrée sont : les paquets du réseau, les fichiers logs et les traces des appels système. Les sondes rassemblent et envoient les informations collectées vers un autre module qui est l'analyseur ;
- **Les analyseurs** : Les analyseurs reçoivent comme entrée les données qui proviennent d'un ou plusieurs collecteurs ou à partir d'autres analyseurs qui auraient au préalables procédé à un premier traitement des données en vue de les raffiner et faciliter la tache à l'analyseur final. L'analyseur peut être considéré comme le noyau de l'IDS, il est responsable de déterminer si une intrusion s'est produite. Comme sortie l'analyseur doit en plus d'un indicateur indiquant une éventuelle attaque, retourner des informations sur l'état du système, et guider l'administrateur en proposant des solutions aux problèmes rencontrés ;
- **Interface utilisateur** : Ce module permet à l'IDS d'interagir avec l'utilisateur, pour pouvoir visualiser les sorties du système, configurer et fixer les paramètres en relation avec la politique de sécurité.

4.3 CLASSIFICATION DES IDS

La classification des systèmes de détection d'intrusion est un sujet difficile à cerner. La raison principale est que la plupart des IDS sont basés sur plus d'une approche et peuvent implémenter un grand nombre de méthodes. Nous présentons dans cette section une typologie des principales méthodes proposées à ce jour. Nous reprenons à cet effet les résultats de laboratoire d'IBM à Zurich [DEB 98].
On doit considérer les cinq caractéristiques suivantes, en premier, lorsqu'on veut examiner un produit IDS :

Le principe de détection utilise,

Le déploiement,

Le comportement en cas d'attaque détectée,

La source des données à analyser,

La fréquence d'utilisation.

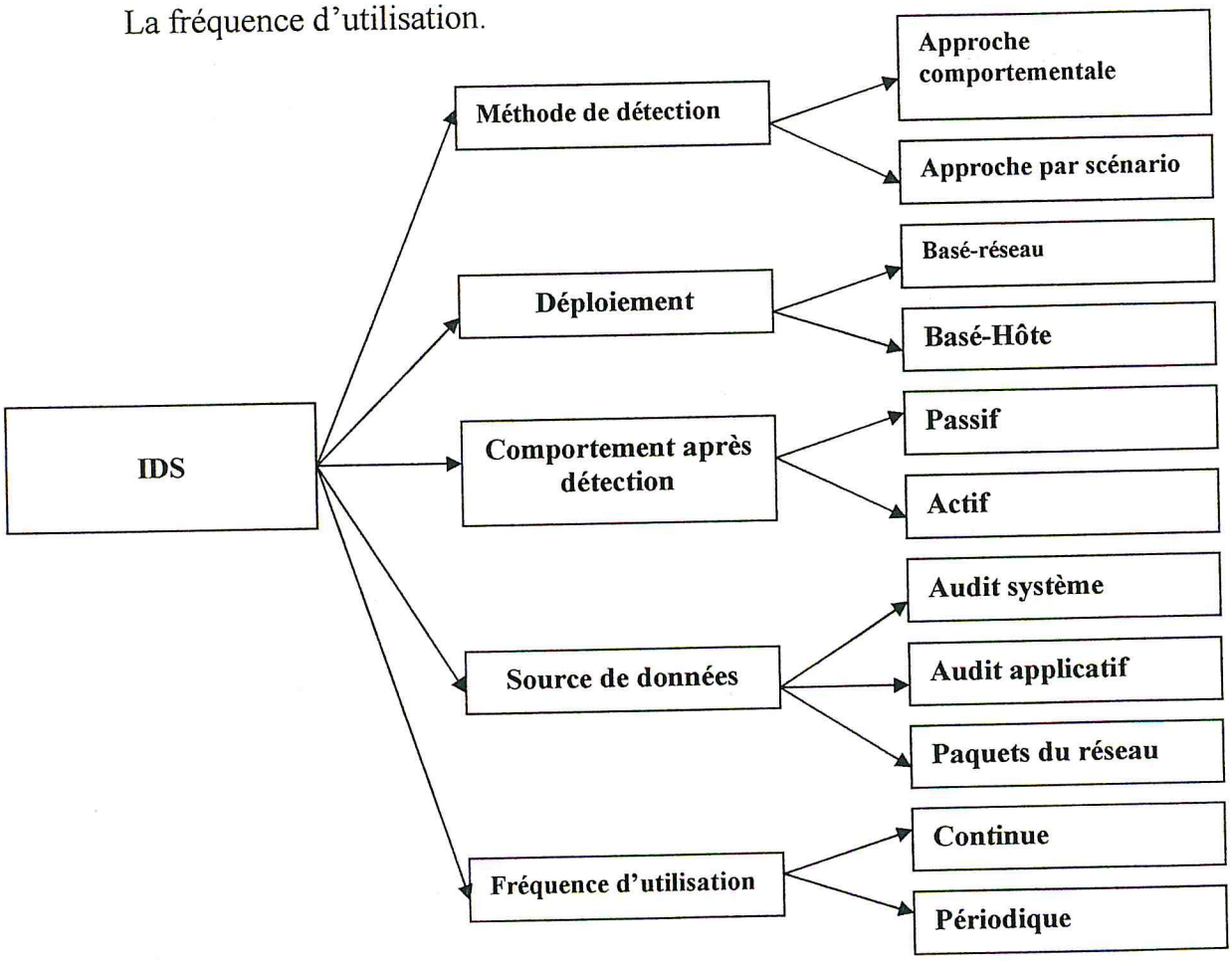


Figure 3. 7: Classification de base des IDS.

4.4 METHODE DE DETECTION

Les deux approches qui ont été proposées à ce jour sont l'approche comportementale et l'approche par scénarios.

4.4.1 APPROCHE COMPORTEMENTALE

Le comportement normal d'un utilisateur ou d'une application (profil) peut être construit de différentes manières. Le système de détection d'intrusions compare l'activité courante au profil. Tout comportement déviant est considéré intrusif. Parmi les méthodes proposées pour construire les profils on cite :

A) Méthodes statistiques : Le profil est calculé à partir de variables considérées comme aléatoires et échantillonnées à intervalles réguliers. Dans un environnement informatique classique (Réseau des machines UNIX et NT), ces variables peuvent être le temps CPU utilisé, la durée et l'heure des connexions, etc. un modèle statistique (ex : covariance) est alors utilisé pour construire la distribution de chaque variable et pour mesurer, au travers d'une grandeur synthétique, le taux de déviation entre un comportement courant et le comportement passé. L'outil NIDES (Next Generation Intrusion Detection Expert Système) utilise entre autres cette méthode;

B) Systèmes experts : Ici, c'est une base de règles qui décrit statistiquement le profil de l'utilisateur au vu de ces précédentes activités. Son comportement courant est comparé aux règles, à la recherche d'une anomalie. La base de règles est rafraîchie régulièrement. L'outil Wisdom & Sense [VAC 89] utilise cette méthode;

C) Graphes : Certaines approches comportementales utilisent des modèles à base de graphes pour mettre en évidence des propriétés et des relations entre ces propriétés. L'intérêt de cette approche est qu'elle permet de traiter plus facilement des événements rares. L'outil GrIDS utilise cette méthode;

D) Réseaux de neurones : La technique consiste à apprendre à un réseau de neurones le comportement normal d'un utilisateur. Par la suite, lorsqu'on lui fournira les actions courantes, il devra décider de leur normalité. L'outil Hyperview [DEB 98] comporte un module de ce type et plusieurs travaux de recherche vont dans le même sens. Cette méthode reste prometteuse, mais n'est pas encore industrialisée ;

E) Immunologie : Cette analogie informatique de l'immunologie biologique a été proposée par Forrest [FOR 97]. Il s'agit de construire un modèle de comportement normal des services réseaux (et non un comportement normal d'utilisateurs) au travers de courtes séquences d'appels système qui sont considérées comme représentatives de l'exécution normale des services considérés. La phase d'apprentissage consiste à observer un service pendant un certain temps afin de construire une base de séquences d'appels normales. En phase de détection, toute séquence étrangère à cet ensemble est considérée comme une potentielle exploitation d'une faille de sécurité du service.

L'approche comportementale permet de détecter des attaques inconnues auparavant ainsi que les abus de privilèges des utilisateurs légitimes du système ; par contre, le comportement de référence n'étant jamais exhaustif, on s'expose à des risques des fausses alarmes (faux positifs). De plus, si des attaques ont été commises durant la

phase d'apprentissage, elles seront considérées comme normales (risque de faux négatifs).

4.4.2 L'APPROCHE PAR SCENARIOS

Des scénarios d'attaques sont construits et l'analyse des traces d'audit se fait à la recherche de ces scénarios. Parmi les méthodes proposées à ce jour à cet effet nous citons :

A) Les Systèmes Experts : Le système expert comporte une base de règles qui décrit les attaques, les événements d'audit sont traduits en des faits qui ont une signification sémantique pour le Système Expert. Son moteur d'inférence décide alors si une attaque répertoriée s'est ou non produite ;

B) Les algorithmes génétiques : Ils sont utilisés pour rechercher des attaques dans des traces d'audit. Chaque individu de la population code un sous-ensemble particulier d'attaques qui sont potentiellement présentes dans les traces d'audit. La valeur d'un individu est proportionnelle au degré de réalisme de l'hypothèse qu'il code, au vu du fichier d'audit ;

C) Pattern matching : Il s'agit là de la méthode la plus en vue actuellement. Des signatures d'attaques sont fournies, à des niveaux sémantiques divers selon les outils (de la suite d'appels système aux commandes passées par utilisateur). Divers algorithmes sont utilisés pour localiser ces signatures dans les traces d'audit (voir [DEB 98] pour un exemple). Les outils Realsecure ou Netranger utilisent cette méthode.

4.5 DEPLOIEMENT

Il y a deux stratégies fondamentales de déploiement pour un produit IDS, basé-Hôte (hoste-based) et basé-réseau (network-based). Le placement d'un produit IDS sur le réseau détermine directement le type d'information qui peut être assemblée et analysée par l'IDS. Le placement des IDS va dépendre de la politique de sécurité menée. Mais il serait intéressant de placer des IDS :

- dans la zone démilitarisée (attaques contre les systèmes publics),
- dans le (ou les) réseau privé (intrusions vers ou depuis le réseau interne),
- sur la patte extérieure du firewall (détection de signes d'attaques parmi tout le trafic entrant et sortant, avant que n'importe quelle protection intervienne).

4.5.1 IDS BASE-RESEAU (NIDS)

Un NIDS évalue l'information capturée à partir des réseaux de communications. Il analyse les trames de paquets voyageant à travers le réseau, et peut aussi faire une analyse du trafic. Les paquets sont habituellement analysés par rapport aux attaques définies en fonction de leur contexte et de leur contenu.

Un NIDS comprend un software qui est installé sur des stations de travail dédiées placés à des emplacements critiques du réseau (ex : à l'extérieur du Firewall, devant un serveur Web ou devant un serveur de messagerie). Celui ci «sniff» c'est à dire qu'il capture et lit les trames qui traversent le réseau. La figure 3.8 montre une topologie de réseau simplifiée, où un moniteur passif a été déployé.

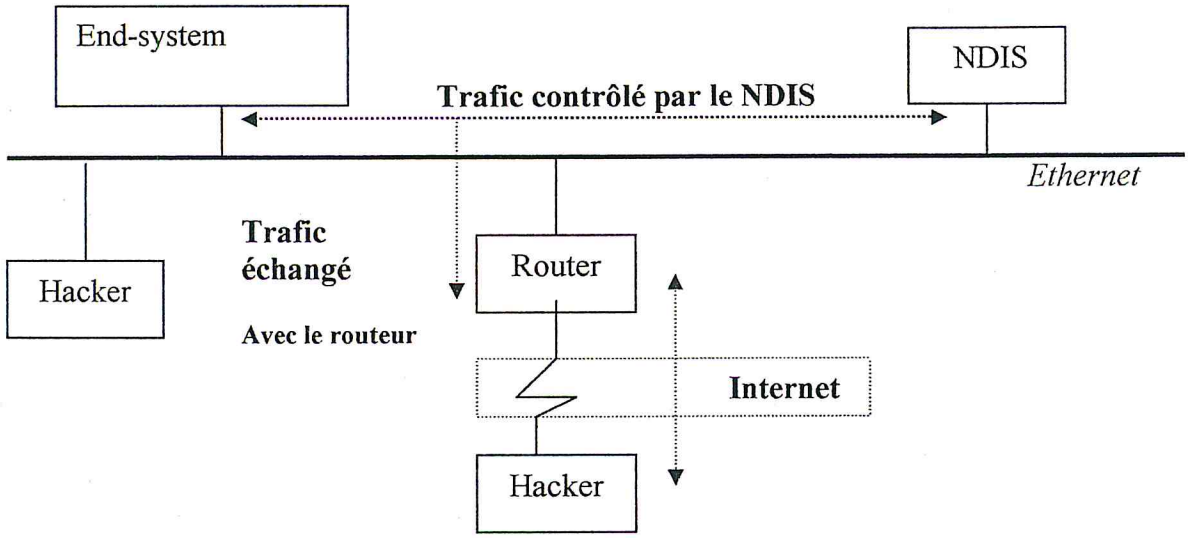


Figure 3. 8: Exemple d'un réseau utilisant un NIDS.

4.5.2 IDS BASE-HOTE (HIDS)

Les IDS Systèmes 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. Il faut cependant noter qu'ils sont incapables de détecter les attaques exploitant les faiblesses de la pile IP du système, typiquement les Dénis de service comme SYN FLOOD ou autre.

4.6 COMPORTEMENT EN CAS D'ATTAQUE DETECTEE

Le comportement d'un outil de détection d'intrusion face à la suspicion d'attaque qu'il est en mesure de révéler est une fonctionnalité qui peut s'avérer importante dans le choix d'un IDS, il peut se contenter de déclencher une alarme (réponse passive), ou prendre des mesures visant à stopper une attaque (réponse active).

4.6.1 REPONSE PASSIVE

C'est le minimum que l'on puisse attendre d'un IDS en matière de comportement en cas d'attaque détectée, la majorité des produits apportent une réponse passive à l'intrusion. Cette réponse se traduit par la génération d'une alarme sous forme de :

- Message sur la console système,
- Message syslog,
- Courrier électronique, voir même alerte sur un beeper,
- Génération de rapport.

De toutes les manières, c'est à l'administrateur système de prendre les mesures qui s'imposent face à la menace d'intrusion révélée.

4.6.2 REPONSE ACTIVE

Les outils commerciaux récents apportent de plus en plus de réponses actives à la détection d'une intrusion. Grâce à une inter-operabilité grandissante avec d'autres produits de sécurité (Firewall, routeurs), les IDS sont en mesure d'offrir automatiquement des contre-réactions aux attaques. Ces mesures réactives sont notamment :

–Fermeture de sessions suspectes,

–Reconfiguration «à la volée » d'un routeur et (ou) d'un Firewall afin de filtrer un flux suspect ;

Cette caractéristique apporte une solution efficace pour se prémunir des dangers que représentent les attaques en déni du service.

4.6.3 IDS A REACTION ABUSIVE

Dans certaines circonstances, l'IDS peut devenir un instrument de déni de service. Les contre-mesures employées peuvent être renversées pour bloquer complètement un trafic légitime ou pour fermer les connexions valides.

4.7 SOURCES DES DONNEES A ANALYSER

Les sources de données à analyser sont une caractéristique essentielle des systèmes de détection d'intrusions. Les données proviennent, soit de fichiers générés par le système d'exploitation, soit de fichiers générés par des applications, soit encore d'informations obtenues en écoutant le trafic sur le réseau.

4.7.1 SOURCES D'INFORMATION SYSTEME (AUDIT SYSTEME)

Un système d'exploitation propose plusieurs sources d'information :

–**Historique des commandes systèmes :** tous les systèmes d'exploitation fournissent des commandes pour avoir un « instantané » de ce qui se passe. Ainsi, sous

UNIX, des commandes telles que ps, pstat ou vmstat fournissent des Informations précises sur les événements système.

–**Accounting :** l'accounting fournit de l'information sur l'usage des ressources partagées par les utilisateurs (temps CPU, mémoire, espace disque,...) les modules statistiques et neuronaux [DEB 98] utilise cette source d'information.

–**Système d'audit de sécurité :** tous les systèmes d'exploitation proposent ce service pour définir des événements, les associer à des utilisateurs et assurer leurs collecte dans un fichier d'audit. On peut donc potentiellement disposer d'informations sur tout ce que font les utilisateurs : accès en lecture à un fichier, exécution d'une application... cette source est utilisée, par exemple, par Gatassa [MEb 98].

4.7.2 SOURCES D'INFORMATION APPLICATIVES :

Les grandes catégories d'applications savent toutes générer des informations sur l'utilisation qui en est faite. C'est le cas des fichiers de logs générés par les serveurs FTP et les serveurs Web. Peu de systèmes de détection d'intrusions les utilisent. On peut toute fois citer l'outil webstakler.

4.7.3 SOURCES D'INFORMATION RESEAU

Des dispositifs matériels ou logiciels (sniffers) permettent de capturer le trafic réseau. Cette source d'information est intéressante car elle permet de rechercher les attaques en déni de service qui se passent au niveau réseau et les tentatives de pénétration à distance. Néanmoins, il est difficile de savoir qui est à l'origine de l'attaque car il est facile de masquer son identité en utilisant le *spoofing*. Presque tous les outils (commerciaux) récents utilisent cette source d'information.

4.8 FREQUENCE D'UTILISATION

La dernière caractéristique des IDS est leur fréquence d'utilisation : périodique ou continue.

4.8.1 SURVEILLANCE PERIODIQUE

Dans un contexte peu sensible, on peu se contenter d'une analyse périodique des fichiers d'audit à la recherche d'éventuelles intrusions passées. On peut, par exemple, choisir une analyse journalière, programmée la nuit afin de ne pas gêner les utilisateurs.

4.8.2 SURVEILLANCE CONTINUE

La tendance actuelle du marché commercial est des produits de détection d'intrusion est de fournir des capacités de surveillance continue, en quasi-temps réel, notamment pour l'analyse des paquets réseaux (NIDS). Dans un contexte sensible ce mode de fonctionnement est en effet justifié. Mais dans certains cas, le dimensionnement du

système devra être effectue en conséquence, car le coût d'un calcul d'une analyse à la volée est loin d'être négligeable.

4.9 LIMITATIONS DES IDS EXISTANTS

Beaucoup d'IDS existants ("network-based" et "host-based") réalisent leur collecte de données ainsi que leur analyse en utilisant une architecture monolithique. C'est à dire la donnée est collecté par une unique station soit à partir d'audit *trails* soit par supervision des paquets du réseau et qu'elle est ensuite analysée par un seul module utilisant différentes techniques. Les autres IDS réalisent la collecte des données (ainsi que certains pré calculs) sur des modules incorporés dans les stations supervisées. Toutefois, la donnée reste envoyée à une entité centrale ou elle est analysée par un moteur monolithique.

Il existe un certain nombre de problèmes issus de ces architectures :

- L'analyseur central est un point de faille à lui seul. Si un intrus parvient à le faire tomber, la totalité du réseau se retrouve sans protection ;
- L'élasticité du réseau est limitée. Le calcul de toutes les informations sur une seule station implique des limites sur la taille du réseau à observer. Au-delà de cette limite, l'analyseur central devient incapable de gérer le flot d'informations. La collecte de données peut également engendrer des problèmes lors de trafics excessifs sur le réseau ;
- Il est difficile de reconfigurer ou d'ajouter des possibilités à un IDS. Les changements et mises à jour sont habituellement effectués en éditant un fichier de configuration et cela en ajoutant une entrée dans une table ou en installant un nouveau module. L'IDS nécessite habituellement d'être redémarré afin de prendre en compte ces changements ;
- L'analyse des données du réseau peut être imparfaite. Réaliser la collecte de données d'un réseau ailleurs que sur la station destinée à les recevoir peut offrir à des intrus la possibilité d'attaques dites d'insertion ou d'évasion. Ceux-ci se servent des failles dans les piles de protocoles du réseau de différents centres serveurs pour dissimuler des attaques ou des dénis de service.

Il reste encore du chemin avant que cette technologie n'arrive à maturité. De nouveaux axes de recherches dans le domaine émergent également. On peut citer notamment les technologies multi-agents, qui visent à répondre au problème de la détection d'intrusions en environnement hétérogène distribué (MAIDS - *Multi Agent for Intrusion Detection System*).

5. CONCLUSION

Nous avons succinctement présenté ici les différents outils de sécurité réseaux existants. Ce marché est, à l'heure actuelle, en plein essor. Evaluer un produit de ce type est une chose extrêmement difficile car la plupart du temps, ce sont des produits fermés (du moins pour ce qui est des outils commerciaux).

De plus, il ne faut pas non plus oublier que le niveau de technicité et de complexité des attaques systèmes et réseaux évolue lui aussi, et impose aux produits de sécurité d'être toujours plus puissants et plus complets.

Après avoir étudier les déférents systèmes de sécurité et leurs inconvénients, on a vue que les IDS ont l'avantage de détecter les attaques à l'intérieur et à l'extérieur de réseau. Pour le développement de ce IDS plusieurs possibilités ont été posées, nous avons choisi celles qui nous offriront - plus tard - une implémentation meilleure et plus flexible. On adopter alors l'approche Pattern matching.

Dans le chapitre suivant nous présenterons les différentes étapes pour réaliser notre scanneur de sécurité, qui est un NIDS a réaction passive.

CHAPITRE IV

CONCEPTION ET MISE EN ŒUVRE

1. INTRODUCTON

Dans le chapitre précédent on a détaillé les déférentes solutions pour améliorer la sécurité dans un réseau, cependant la solution des IDS c'était notre objective et pour cela on a suivis les étapes qui seront expliqué dans ce chapitre.

1.1 METHODE DE DEVELOPPEMENT

Dans la phase de développement de ce projet on a choisie une méthode cascade conforme à celle du modèle utilisé dans la norme AFNOR Z 67-150[PRI 98]. Ce modèle comporte sept phases ou activités principales que nous allons détailler brièvement dans la figure 4.1.

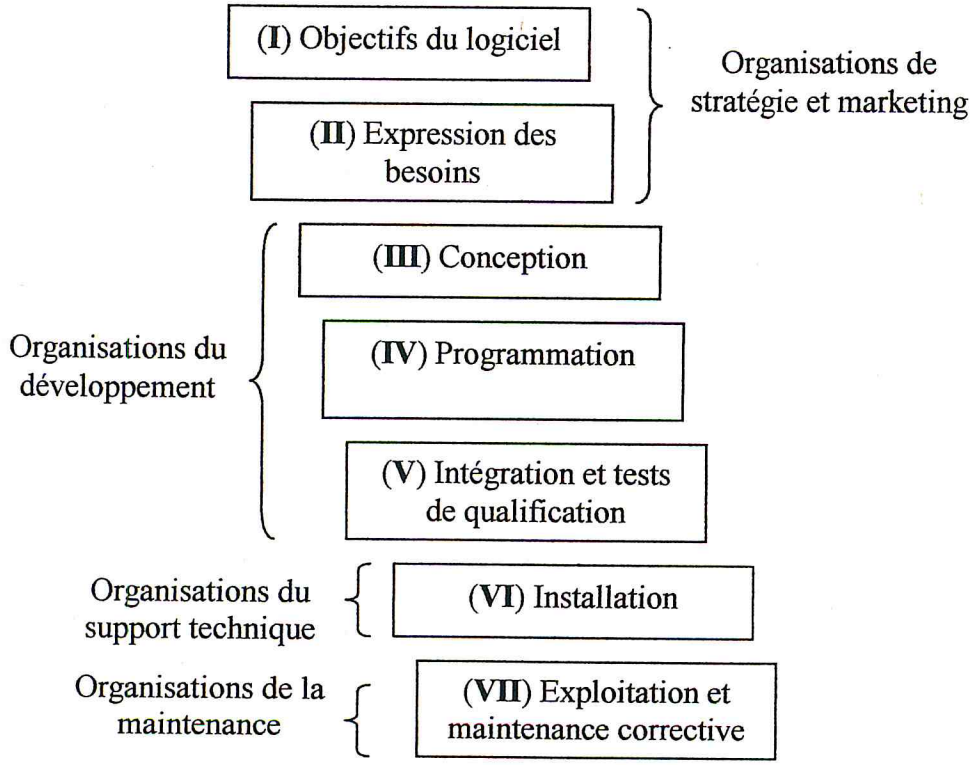


Figure 4. 1 : phases de développement de projet

Phase I objectifs du logiciel : cette phase initiale du développement de tout projet logiciel donne une description et une évaluation globale des besoins que notre logiciel est censé satisfaire :

- analyse de réseau,
- évaluation des risques.

Phase II expression des besoins : on décrit dans cette phase les fonction que le logiciel doit effectuer :

- Détecter les ports ouverts (scanner des ports);
- Chercher des failles dans le réseau ;
- Analyser le trafic pour chercher les paquets suspects.

Phase III conception: cette phase à comme objectif de définir de façon très prédise les fonction et l'architecture du logiciel, à partir des besoins exprimés et des contraintes générales définies en I et II, on va détaillé cette phase par la suite.

Phase IV programmation : cette phase correspond à la programmation proprement dite des fonctions sur la base des informations précises venant de la phase de conception, on va détaille cette phase dans le paragraphe « mise en œuvre ».

Phase V intégration et tests de qualification : cette phase correspond au regroupement progressif de tous les modules de façon à garantir la vérification et la validation progressive du logiciel, jusqu'à pouvoir le faire fonctionner dans son environnement réel, pour plus de détail sur les tests voir l'Annexe B.

Phase VI installation : cette phase correspond à la mise en fonctionnement opérationnel du logiciel.

Phase VII exploitation et maintenance : cette phase à pour objet de s'assurer que le logiciel installé fonctionne correctement.

2. L'APPROCHE UTILISE :

On a choisi l'approche par scénario de type NIDS passif basée sur un system pattern mathing, puisque on s'intéresse à des failles qui sont déjà déterminé.

3. PRINCIPE DE FONCTIONNEMENT :

On a utilisé deux principes :

3.1 Un scanner :

On sait bien qu'une attaque à distant de n'importe quel type doit se communiquer avec la victime et puisque les communications en réseau ne peuvent dérouler qu'avec le mécanisme des ports qui sont des portes logiques d'entrer à un ordinateur connecté au réseau, alors c'est claire que si un port est ouvert impliquera que la machine est candidate d'être attaqué surtout si le port ouvert fait partie des port qui sont connue de leur utilisation par les intruse.

Notre scanner scanne tous les port des machines connecté au réseau locale et donnera une cartographie de ces machine à savoir pour chaque adresse ip détecte l'état des port et ainsi les service exécutés sur les ports ouverts.

Ce scanner peut être l'objet d'un **FDS** (failles détection system)ou un system de détection des faille ,en effet le scanner nous donne une cartographie du réseaux à savoir les @ip valides ,les ports ouverts ,les services ,l'os(operating system) de chaque machine et son service pack ainsi les déférent serveurs installé dans le réseaux local (serveur Web,messageries, ftp ..) et leur service pack .alors avec ces informations et une base de connaissance qu'elle contient les failles déjà connues ,pour chaque service pack d'un os ou d'un serveur et les failles de certain port , le system peut déceler les failles qu'elle existes en réseau et alertera l'administrateur suite à une faille trouvée

pour agir avant qu'un intruse exploite cette faille pour attaquer le réseau.
L'administrateur peut alors bloquer un port ou mettre à jour un system par des services pack récent …

On peut résumé le scanner par l'organigramme suivent :

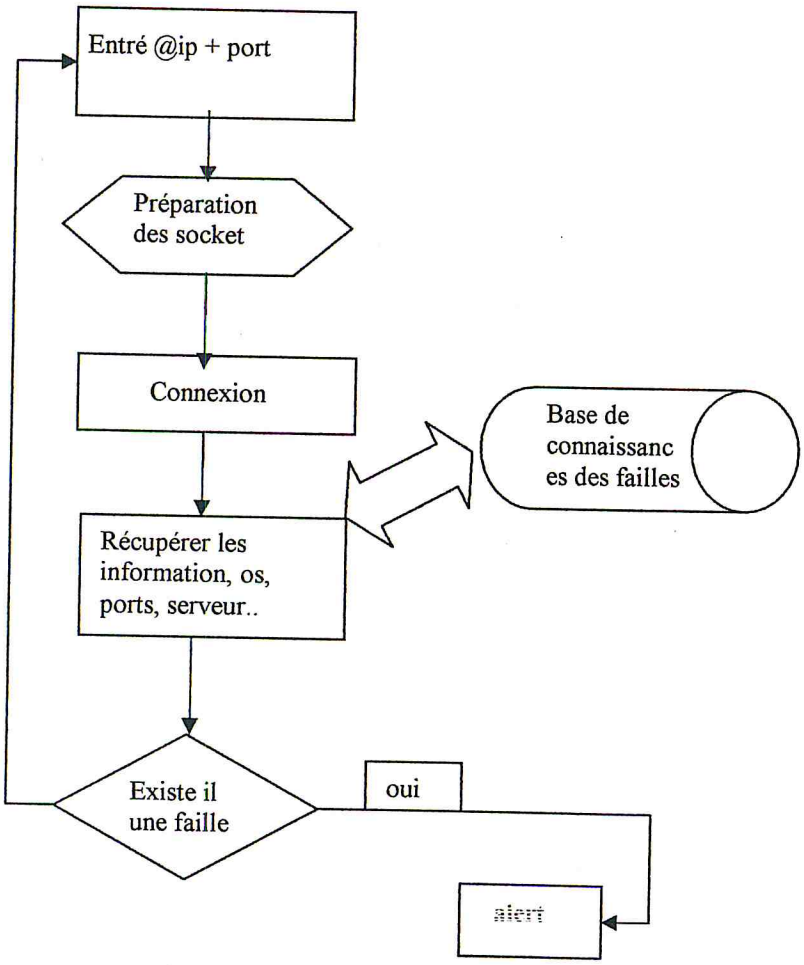


Figure 4.1 : organigramme de fonctionnement du scanner .

3.2 Le system pattern mathing :

Ce module permet de détecter une tentative d'attaque d'où une intrusion à savoir qu'elle est déjà prévue dans la base de connaissance des intrusions.

Pour cela nous somme obligés d'utiliser un snifer qui sert à récupérer les paquet circulent dans le réseau afin de pouvoir effectuer le pattern matching par rapports à la base. On peut résumé le fonctionnement par le diagramme suivent :

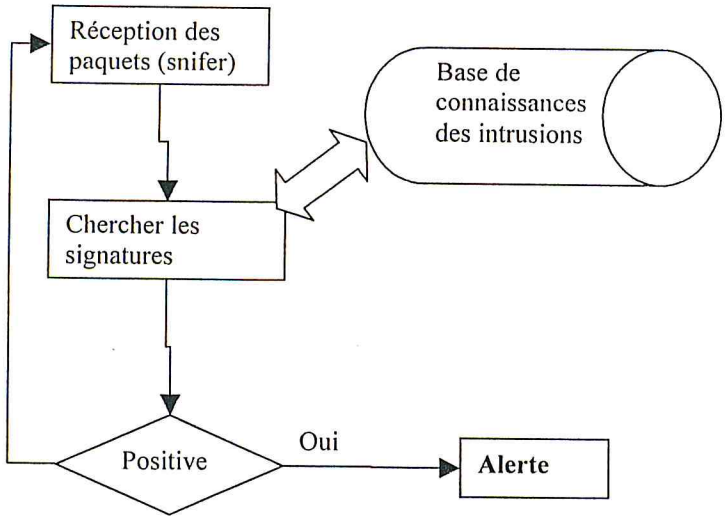


Figure 4. 1: organigramme de fonctionnement du pattern matching.

2.2.1 DESCRIPTIONS DES SIGNATURES

La notation s'inspire des expressions régulières :

{a, b} : veut dire soit a, soit b, [a - b] : n'importe quel élément entre a et b,

&& et || et ! : sont respectivement le ET, OU logiques et NON.

Explicitons tout d'abord le format de sortie des sniffers :

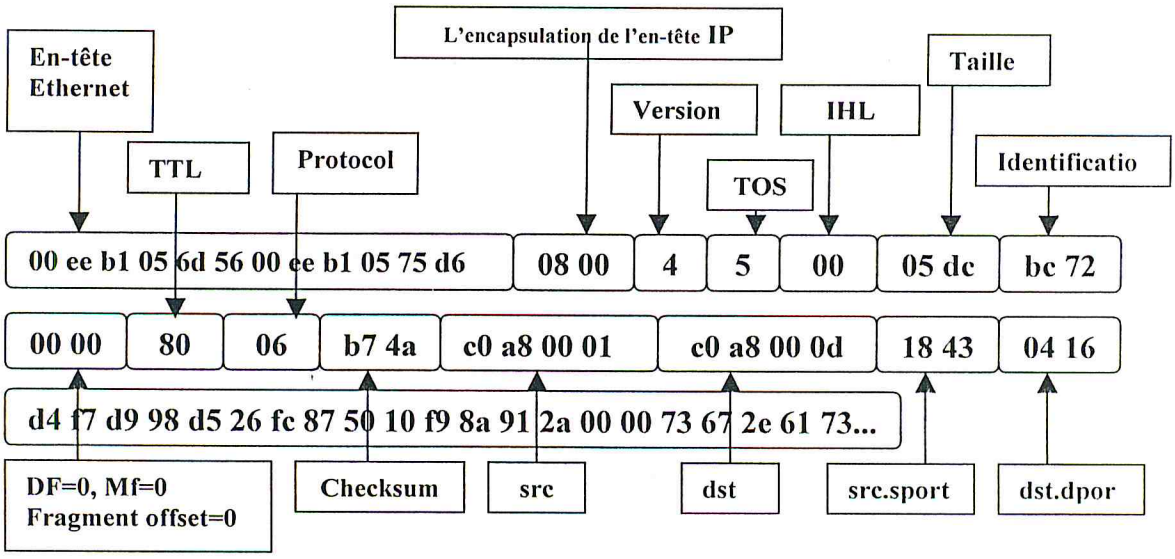


Figure 4. 2 : format de sortie de sniffer

-Pour TCP

Timestamp src.sport > dst.dport: flags data-seqno ack window urgent options Où

Timestamp (TTL) Durée de vie,

dst.dport port de destinataire, **src.sport** port source,

flags est un sous-ensemble non-vide de FLAG

FLAG = {S(SYN), F (FIN), P (PUSH), R (RST), .(NOFLAG:pas de flag activé)},
data-seqno est de la forme Start-SN :End-SN(End-SN - Start-SN),
ack Vaut soit ACK ack-num ou est vide (¤),
Window la valeur de la fenêtre,
Idem pour urgent : URG ou vide (¤),
L'option de fragmentation est la principale option qui nous intéresse :
(frag ID :size@offset{+, ¤}) (Le + indique que le paquet n'est pas complet)
L'en-tête TCP contient des champs (réservés), pour une éventuelle mise à niveau du protocole. Si les bits correspondants sont activés, le sniffer renvoie un message d'erreur (le paquet TCP est non valide).

-Pour UDP

Timestamp src.sport > dst.dport: udp data

-Pour ICMP

Timestamp src > dst : icmp : message

src : adresse IP source,
dst : adresse IP destinataire,
Les champs laissés par défaut ou omis peuvent prendre n'importe quelle valeur. On omettra en particulier le champ timestamp pour les attaques ponctuelles.

3.2.2 SIGNATURES DES ATTAQUES PONCTUELLES

Les attaques dites ponctuelles sont celles détectables sans problème sur une seule ligne dans la base de connaissance des intrusions. La plupart de ces attaques consistent en un seul paquet très particulier (au sens où il est aisément identifiable), les autres sont basées sur la répétition (en vue d'une *inondation*) d'un paquet anormal, ce qui rend par conséquent leur détection possible sur un seul paquet.

A) attaque TCP

-LAND

L'attaque consiste à envoyer un paquet où la source et la destination sont les mêmes : certains systèmes d'exploitation ne savent pas gérer ce type de paquets et on assiste à un "plantage".

src.sport > src.dport

-BROADCAST TCP

TCP est un protocole orienté connexion, un broadcast n'a pas lieu d'être.

src.sport > [0 - 255]. [0 - 255]. [0- 255]. {0,255}.dport

-ATTAQUES SUR PORTS

Tentative d'atteindre des ports interdits ou suspects : 0, ports typiques utilisés par les chevaux de Troie ... Cette signature est parmi les plus vulnérables

src.sport > dst. {0, 31337, ...}

- WinNUKE

Cette attaque provoque un écran bleu sur les machines Windows : le système ne sait pas gérer les paquets urgents destinés au port netBIOS sous certaines conditions.

src.sport > dst.139: flags data-seqno ack Window URG

- SYN + DATA

Il peut s'agir d'une tentative d'échapper à la détection : il n'est pas prévu que des données circulent lors du handshake. Certains IDS ne s'occupant pas des paquets contenant des données, ces paquets anormaux passent à travers les filtres.

Src.sport > dst.dport : S Start-SN : !Start-SN

(ie End-SN différent de Start-SN)

B) attaque UDP

- ECHO-CHARGEN :

Cette attaque revient à établir une boucle infernale faisant converser indéfiniment les ports echo (service renvoyant les caractères qu'on lui présente en entrée) et chargen (service générateur de caractères aléatoires). Ce genre de trafic étant hautement inhabituel, l'attaque est détectable avec un seul paquet, même si elle s'apparente à une attaque par flooding donc temporelle.

(Src.7 > dst.19 || src.19 > dst.7) : udp

- FRAGGLE-AMPLI

L'attaque fraggle (de même que l'attaque smurf, voir après) permet d'utiliser de grosses ressources pour frapper la victime, sans pour autant les posséder. L'idée sous-jacente est d'envoyer une requête quelconque au nom de la victime auprès d'un réseau choisi qui servira d'amplificateur. Ainsi, la victime recevra les réponses (non sollicitées) d'un réseau entier, ce qui peut représenter un grand nombre de paquets à traiter d'un seul coup, entraînant ainsi un Déni de Service. Bien entendu, les paquets d'initiation de l'attaque étant maquillés dès le départ, il est fort improbable que l'attaquant soit retrouvé.

Src.sport > [0- 255]. [0 - 255]. [0 - 255]. {0, 255}. {19,7} : udp

Ici src est la victime, et le réseau visé va faire office d'amplificateur d'attaque. Dans notre cas, la signature peut être allégée en indiquant simplement l'adresse broadcast du réseau protégé.

- FRAGGLE-VICTIME

Cette fois-ci dst est la victime.

Src. {19,7} > dst.dport : udp

Ce pattern sera détecté n = m * r fois, où r est la taille du réseau amplificateur et m le nombre de requêtes envoyé par l'attaquant. On voit bien ici comment le réseau amplificateur permet d'accroître linéairement l'ampleur de l'attaque.

C) attaque ICMP

- WINFREEZE

Cette attaque consiste à envoyer des informations de routage erronées à la victime, notamment en lui faisant croire que la victime est elle-même la prochaine étape sur la route vers la destination voulue. Ainsi, lorsque la victime souhaite contacter cette destination, une boucle se crée avec les conséquences attendues d'un Déni de Service.

Src > dst : icmp : redirect IP to host dst

- SMURF-AMPLI

Il s'agit d'une attaque de type Fraggle, version ICMP.

src > [0 - 255].[0 - 255].[0 - 255].{0,255}:icmp: echo request

3.2.3 SIGNATURES DES ATTAQUES TEMPORELLES

Sous cette désignation, sont regroupées toutes les attaques dont la trace dans la base de connaissance des intrusions est répartie sur plusieurs lignes (ie paquets) parce que les paquets qui composent ces attaques, pris individuellement, sont plus ou moins inoffensifs : les attaques temporelles à proprement parler (scans, balayages et flooding), pour lesquelles existent un seuil de tolérance (nombre de connexions maximum autorisé avant de considérer qu'il s'agit d'une attaque). Et les attaques par fragmentation (attaques impliquant un seul paquet mais découpé en plusieurs morceaux, telles que Teardrop et Ping of Death).

Comment définir ce seuil *?* : Il est assez difficile d'établir un seuil, même d'après l'expérience, puisque cela reste assez subjectif (combien de connexions est-ce que j'autorise avant de considérer qu'il s'agit d'une attaque ?). Pour le flooding, ces seuils varient également en fonction du port visé. Et une fois que ce seuil est défini. *Certains* ***faux négatifs*** *ne peuvent être évités* : un attaquant sera indétectable s'il délaie suffisamment son attaque pour passer en dessous du seuil; ou bien s'il parallélise son scan en utilisant plusieurs machines.

La difficulté est donc de trouver un bon compromis au niveau du seuil afin de limiter le nombre de ces faux négatifs. Malheureusement, on peut être sûr que les scans non

détectés proviennent des adversaires les plus dangereux : ceux qui connaissent les failles des systèmes de détection.

Ici la difficulté, d'un point de vue algorithmique, est qu'il faudrait faire appel à des "meta-patterns" pour modéliser ces attaques. Voici quelques exemples :

- **PORT SCAN** (déterminer les services présents sur une machine)

Time1 src.sport1 >dest.dport1 && (0 ou plusieurs entrées quelconques) &&... && (0 ou plusieurs entrées quelconques) && Timen src.portn > dest.dportn

Avec Timen - Time1 < 1000 ms et dport1 != ... != dportn.

n correspond au nombre maximal de ports "scannables" avant de considérer qu'il s'agit d'une attaque.

- **MITNICK ATTACK** (TCP) (camouflage d'IP)

PORT SCAN sur dport && Time1 src1.sport1 > dest.dport: S && (0 ou plusieurs entrées quelconques) && Time2 src2.sport2 > dest.dport : flags data-seqno ACK

La signature décrit ici l'attaque historique menée par Kevin Mitnick contre le réseau de Tsutomu Shimomura, du point de vue de la machine attaquée (dest). Dans ce cas particulier, il existait une relation de confiance (Rhosts) entre src1 et dest.

On préférera la signature TCP HIJACK pour une détection plus générale des détournements de connexion.

- **STICK** (désactivation à distance de certains NIDS par déni de service)

Time1 src.sport1 >dest.dport R && (0 ou plusieurs entrées quelconques) &&... && (0 ou plusieurs entrées quelconques) && Timen src.portn > dest.dport R Timen - Time1 < 1000 ms. (respect le seuil)

Contrairement aux attaques ponctuelles et assimilées, la détection des attaques temporelles devrait obligatoirement se faire en analysant plusieurs lignes (comme on l'a vu avant, certaines attaques ponctuelles qui sont en fait du flooding sont facilement détectables en raison du port utilisé, etc.). Pour éviter d'utiliser des motifs trop lourds, une alternative intéressante est de faire appel à une structure réactive : chaque fois que nous rencontrons un paquet qui pourrait faire partie d'une attaque, la structure évolue jusqu'à atteindre un niveau critique déclenchant une alerte. Concrètement, ce niveau critique peut être un seuil de caractérisation de flood ou de scan, ou bien une taille totale pour un paquet fragmenté ... Nous associons donc à chaque type d'attaque un objet le décrivant, et une instance sera créée à chaque fois qu'une attaque est

potentiellement en cours, faisant office de mémoire. Ces objets héritent des classes suivantes :

```
classe ICMP
        {
         string icmp_type;
        }
```

```
classe TCP
        {
         list of strings flags;
         // flags est un sous-ensemble de {SYN,FIN,PSH,URG,ACK,RST}ou bien vide
         list of int data-seqno;
         // data-seqno = {Start-SN; End-SN}
         int acknum;
         // si le flag ACK n'est pas activé, acknum = -1
        }
```

```
classe TCP_FRAG héritant de TCP
        {
         int fragID;
         int size;
         int offset;
         boolean more;
         // indique si le bit "More Fragments" est activé
        }
```

Dans la vue d'une utilisation pour la détection d'intrusion par signatures, on ne définis pas de super-classe PAQUET ou UDP. En effet, les objets qu'on utilisera peuvent contenir des listes d'IP ou de ports. On a donc fait ce choix par cohérence et souci de ne pas surcharger les structures ...

Nous avons désormais des classes décrivant la plupart des paquets qu'on aura à filtrer. A partir de là, on peut définir des classes caractérisant les attaques temporelles. Par convention, ces objets sont stockés dans des tables indexées par la caractéristique principale de l'attaque (cible ou attaquant) comme par exemple **attaque_sur_IP[adresse_attaquant]**.+ Ainsi:

- **PING SWEEP** (icmp) :

Cette "attaque" de reconnaissance consiste à bombarder un réseau de requêtes "ping", afin d'obtenir une carte des adresses actives sur le réseau ciblé.

```
classe PINGSWEEP  héritant de ICMP
        {
         string src;
         //à la construction, icmp_type = "echo request"
         time heure_de_debut;
         time heure_en_cours;
         int compteur;
         list of string victims;
         // victims sert à stocker les IPs des machines "pingées"
         function new ...
         function alerte ...
        }
```

```
Timei src > dsti : icmp : echo request
    ==>si !(pingsweep[src])
        {
            pingsweep[src] = new PINGSWEEP
            //  heure_de_debut = Timei et heure_en_cours= Timei;
        }
       si !( desti est dans pingsweep[src].victims )
       {
        pingsweep[src].compteur =+1;
        pingsweep[src].heure_en_cours = Timei;
        ajouter desti à pingsweep[src].victims;
       }
       si ( pingsweep[src].compteur > pingsweep_max &&
pingsweep[src].heure_en_cours - pingsweep[src].heure_de_début < 1000 ms )
        {
        Alerte !
        }
       // pingsweep_max est une variable de seuil à définir
```

- TCP HIJACK

Cette attaque désigne un détournement de connexion selon la méthode présentée précédemment, à l'aide des sequence numbers / acknowledge numbers. Notons que dans ce cas, il est nécessaire de connaître le trafic dans les deux sens. Jusqu'à présent, il était possible de caractériser une attaque par la seule activité de l'agresseur. Ici, pour

déterminer si une attaque est en cours, il faut d'abord vérifier si le numéro de séquence a été volé.

```
        classe HIJACK héritant de TCP
        {
         string victime;
         string spoof;
         time heure_de_debut;
         time heure_en_cours;
        }
Timei src.sport>dst.dport:flags Start-SNi:End-SNi(End-SNi- Start-SNi)ACK ack-numi
   ==>si ( !hijack[src.sport] )
        {
         //src : machine à protéger, on pourrait donc éventuellement filtrer les machines
         // pour lesquelles on crée cet objet
         hijack [src.sport] = new HIJACK;
         // hijack.victime=src.sport, hijack.spoof=dst.dport, hijack.acknum = ack-numi,
         //hijack.data-seqno[1]=End-SNi, hijack.data-seqno[0]=Start-SNi,
         hijack.heure_de_debut = Timei;
        }
        si ( hijack[src.sport] )
        {
         hijack.acknum = ack-numi;
         hijack.data-seqno[1] = End-SNi;
         hijak.heure_de_debut = Timei;
         //mise à jour des paramètres cruciaux
        }
        si ( hijack[dst.dport] )
        {
         // src est l'attaquant potentiel, ce coup-ci
        si ( hijack.spoof != src.sport && Start-SNi == hijack.acknum && ack-numi ==
hijack.dta-seqno[1] )
         {
              hijack.heure_en_cours = Timei;
              Alerte !
              // Les 3 conditions du spoofing sont réunies
         }
        }
```

Cette signature est peut-être moins évidente à saisir que les autres : supposons que nous avons ce paquet :

```
Timei victime.sport > spoof.port: flags Start-SNi:End-SNi(End-SNi - Start-SNi) ACK ack-numi
```

Victime envoie un paquet à la machine spoof, qui peut potentiellement servir de masque à un attaquant. Le paquet normalement attendu si la machine spoof existe bien et a effectivement sollicité le paquet précédent est de la forme :

```
Timej spoof.sport > victim.dport: flags ack-numi:End-SNj(End-SNj - ack-numi) ACK End-SNj
```

Par contre, en cas de détournement de connexion avéré, nous verrions passer un paquet de cette sorte :

```
Timej attaquant.sport > victim.dport : flags ack-numi:End-SNj(End-SNj-ack-numi) ACK End-SNj
```

Où l'attaquant est une machine différente de spoof. Cependant, pour confirmer le détournement de connexion, il faut vérifier les numéros de séquence et d'acquittement, formant un doublet unique caractérisant l'avancement d'une connexion (en effet, un même port d'une machine peut être accédé par plusieurs machines en même temps, comme le port http par exemple) tout comme le quadruplet {src,sport,dst,dport} caractérise de façon unique une connexion . Si une machine usurpe un tel doublet, c'est qu'elle tente de détourner la connexion.

- SYN FLOOD (TCP)

Lorsque le port d'une machine est sollicité pour une connexion TCP, la machine garde une trace de ce contact dans la *queue de connexion (connection stack)*. Ainsi, lorsque le paquet d'acquittement revient (la 3e partie du handshake), la machine sait que tout va bien. La trace est maintenue dans la queue un certain temps qui dépend des systèmes d'exploitations, et si l'acquittement n'est pas arrivé avant cette limite temporelle la connexion est considéré comme perdue et la trace est retirée de la queue. Or cette queue a bien évidemment une capacité limitée. L'idée est donc de bourrer cette queue de requêtes de connexions, si bien que toute connexion légitime ne pourra être stockée dans la queue et donc aboutir .De l'extérieur, le port de la victime semble inactif. Ici deux patterns agissent conjointement :

```
classe SYNFLOOD
        {
        time heure_de_debut;
        time heure_en_cours;
        string dst;
        int dport;
        list of string attackers;
        // contient les adresses IP des éventuels attaquants
       }
```

```
Timei srci.sporti > dst.dport : S
==>    si !(synflood[dst.dport])
        {
       créer synflood[dst.dport]
       // heure_de_debut = Timei et heure_en_cours= Timei;
        }
       ajouter srci.sporti à synflood[dst.dport].attackers;
       synflood[dst.dport].compteur =+1;
       si ( synflood[dest.dport].compteur > synflood_max[dport] )
        {
        Alerte !
        }
```

```
Timei srci.sporti > dst.dport : flags data-seqno ACK ack-num
==> si !(synflood[dst.dport])
      {
      Alerte !
      //un paquet ACK sans handshake préalable n'est pas normal
      //( soit un détournement de connexion, soit une tentative d'échapper aux IDS )
      }
      sinon
      {
      retirer srci.sporti de synflood[dst.dport].attackers;
      synflood[dst.dport].compteur --;
      }
```

Juste avant que la file d'attente soit engorgée (à paramétrer avec synflood_max[dport], qui dépendra du port visé), l'alerte est donnée . Il faut pouvoir tenir compte du temps, afin de ne plus tenir compte des connexions qui ont été automatiquement désactivées par Timeout. D'autre part, on parvient à détecter une attaque que le pattern matching simple ne pouvait pas relever (nécessité d'un retour en arrière) : le détournement du 3-way handshake de TCP.

- PORT SCAN

Le scanning de port sert à déterminer les services présents sur une machine.

```
classe PORTSCAN héritant de TCP_FRAG
   {
      TCP_FRAG
      {
```

```
        string dest;
        string src;
        string type;
        list of int ports; // ports scannés
              int compteur;
      }
  }
```

```
Time1 src.sporti > dest.dporti : flags data-seqno ack window urg options
      si !(portscan[dest])
      {
       si ( flags == SYN )
       {
       portscan[dest] = new PORTSCAN
       // heure_de_debut = Timei et heure_en_cours= Timei;
       portscan(src).type = "SYN scan";
       }
       sinon
        si ( flags == FIN )
        {
        portscan[dest] = new PORTSCAN
        // heure_de_debut = Timei et heure_en_cours= Timei;
        portscan(dest).type = "FIN scan";
        }
       sinon
        si ( flags == "" && ack == ¤ && urg == ¤ )
        {
       portscan[dest] = new PORTSCAN
       // heure_de_debut = Timei et heure_en_cours= Timei;
       portscan(dest).type = "NULL scan";
       }
       sinon
        si ( flags == "FIN,PUSH" && urg == "URG" )
        {
        portscan[dest] = new PORTSCAN
        // heure_de_debut = Timei et heure_en_cours= Timei;
        portscan(dest).type = "XMAS scan";
        }
        si ( sporti == 20 )
```

```
        {
        portscan[dest].type .= " via FTP bouncing";
        }
        si ( options contient  (frag ID :size@offset{+, ¤}) )
        {
        // paquet fragmentés pour tromper les IDS
        portscan[dest].type .= " avec fragmentation";
      }
      }
      si ( dporti n'est pas dans portscan[dest].ports )
      {
      ajouter dporti à portscan[dest].ports;
      portscan[dest].compteur =+1;
      }
      si ( portscan[dest].compteur > scan_max )
      {
      //scan_max =10 parait être un seuil correct
      //surtout si on ne tient pas compte du temps
      Alerte !
      }
```

- STICK

Stick est une tentative de désactivation à distance des systèmes de détection d'intrusion orientés réseau. L'attaque consiste à envoyer un très grand nombre de paquets RST (fin de connexion brutale), ce qui a pour effet de surcharger de travail le NIDS. Ceci aura éventuellement pour effet de provoquer un déni de service contre le système de détection d'intrusion, laissant le réseau protégé sans surveillance.

```
classe STICK héritant de TCP
        {
        string dest;
        string src;
        time heure_de_debut;
        time heure_de_fin;
        int compteur;
        }
```

```
Timei src.sporti > dest.dport : R =>
      si !(stick[dest.dport])
      {
```

```
        stick[dest.dport] = new STICK
        // heure_de_debut = Timei et heure_en_cours= Timei;
        }
        stick[dest.dport].compteur =+1;
        si ( stick[dest.dport].compteur > stick_max )
        {
//stick_max=10? Plus de 10 paquets RST pour fermer une connexion de façon abrupte!
        Alerte ! }
```

- **SMURF-VICTIME** (ICMP)

Principe identique à fraggle-victime.

```
classe SMURF_VICTIME héritant de ICMP
        {
        string dest;
        list of string attackers;
        // stocke les IP des machines amplificatrices
        time heure_de_debut;
        time heure_en_cours;
        int compteur;
        }
```

```
timei srci > dst : icmp : echo reply ==>
        si !(smurf-victim[dst])
        {
        smurf-victim[dst] = new SMURF_VICTIME;
        // heure_de_debut = Timei et heure_en_cours= Timei;
        }
        ajouter srci à smurf-victim[dst].attackers;
        smurf-victim[dst].compteur =+1;
        si ( smurf-victim[dst].compteur > smurf_max )
        {
        Alerte !
        }
```

2.2.4 SIGNATURE DES ATTAQUES PAR FRAGMENTATION

Le concept de fragmentation

La fragmentation a lieu lorsqu'un datagramme IP en transit doit passer par un réseau dont la *taille maximale de transmission* (MTU) est plus petite que la taille du datagramme (en clair, il y a engorgement). Par exemple, la MTU d'Ethernet est de 1500 octets; donc un datagramme de taille supérieure à 1500 octets devra être fragmenté pour voyager sur Ethernet. Les fragments se comportent exactement comme des paquets normaux, si ce n'est qu'ils sont réassemblés par la machine destinataire. Pour se faire, chaque fragment contient les informations suivantes :

- ***Frag ID*** identifiant de fragment, permettant à la machine destination de regrouper tous les fragments appartenant à un même paquet,
- **un *offset*** indiquant la place du fragment dans le paquet original,
- **la taille** des données contenues dans le paquet fragmenté,
- **un indicateur** pour savoir si le paquet fragmenter est suivi par d'autres paquets ou bien si il s'agit du dernier fragment. Cet indicateur est le flag more fragments (MF).

Toutes ces informations se situent dans l'en-tête IP, cet en-tête étant lui-même suivi par un fragment encapsulé (Voir chapitre 2).

Les traces de paquets fragmentés sont un peu différentes de celles des paquets normaux, et reflètent l'absence d'informations fournies par l'en-tête TCP. Dans ces conditions, les fragments ne contenant pas l'en-tête ont ce format générique :

Timestamp src > dst : (frag ID : size@offset {+, ¤})

Les attaques suivantes fonctionneront donc avec deux signatures : une première pour détecter le fragment contenant l'entête, et la suivante pour repérer les informations caractéristiques de l'attaque par fragmentation proprement dite.

- TEARDROP (TCP)

L'attaque exploite le chevauchement de paquets fragmentés.

Ici, une difficulté supplémentaire est à prendre en compte : les paquets n'arrivent pas forcément dans le bon ordre (dans l'ordre des offsets). Pour un paquet fragmenté normal, et pour toute valeur de i, nous savons que

offset_i + size_i = somme (size_j, j = 1 ... i).

Par conséquent, une attaque teardrop est telle qu'il existe une valeur de i pour laquelle

offset_i + size_i < somme(size_j, j = 1 ... i) .

```
classe TEARDROP héritant de TCP_FRAG
        {
        string dst;
        int dport;
        string src;
        int sport;
        int ID;
        // ID number du paquet fragmenté
```

```
        int offset_max;
        // le plus grand offset reçu en cours
        int taille_theorique;
        // egal à offset + size du dernier fragment reçu
        int taille_reelle;
        // somme des size_i reçus
        }
Timei SRC.SPORT>DST.DPORT:flags data-seqno ack window(frag ID:size_i
@offset_i{+,¤})
==>si !( teardrop[ID] )
        {
        teardrop[ID] = new TEARDROP;
        }
        teardrop[ID].dst = DST;
        // etc ...
```

et

```
Timei SRC > DST : (frag ID :size_i@offset_i{+,¤})
==>si ( offset_i == max ( offset_i, teardrop[ID].offset_max) )
        {
        teardrop[ID].offset_max = offset_i ;
        teardrop[ID].taille_theorique = offset_i + size_i;
        }
        teardrop[ID].taille_reelle =+ size_i;
        si ( teardrop[ID].taille_theorique < teardrop[ID].taille_reelle )
        {
        Alerte !
        //la condition ci-dessus n'est pas nécessaire mais elle est suffisante
        //elle devient nécessaire lorsque tous les fragments sont arrivés
        //A défaut de trouver mieux ...
        }
```

- **PING OF DEATH** (ICMP)

L'envoie d'une requête ping trop grosse (de taille supérieure à 65 ko) provoque un plantage de la machine cible, et donc un déni de service.

```
classe POD héritant de ICMP
        {
        string src;
        string dst;
        int ID;
```

```
        int total_size;
        }
```

```
Timei src > dst : icmp : echo request (frag ID :size_i@offset_i{+,¤}) ==>
        si !(pod[ID])
        {
        pod[ID] = new POD;
        }
et Timei SRC > DST : (frag ID:size_final@offset_final¤) ==>
        // si pod[ID] existe+
        pod[ID].total_size = offset_final + size_final;
        si ( pod[ID].total_size > 65 k )
        {
        Alerte !
        }
```

La taille du paquet Ping est connue grâce au dernier paquet de la fragmentation (celui ne contenant pas de +), il suffit donc de repérer celui-ci, et de tester la taille du paquet. Une fois ce test effectué, on peut détruire l'objet pod associé dans la table.
Sur Teardrop, l'inconvénient est qu'on n'a pas de moyen de détruire l'objet une fois que le paquet complet a été reçu, puisque les paquets fragmentés arrivent dans le désordre. Ces attaques restent assez rares, ce qui devrait limiter le nombre d'objets correspondants créés.

4-LA mise en œuvre de notre conception

Dans cette étape finale on a essayer d' implémenter le maximum de module afin d' aboutir à un résultat efficace et qu' il répond à la conception proposé .

4.1 Le model type qu' on veut implémenter :

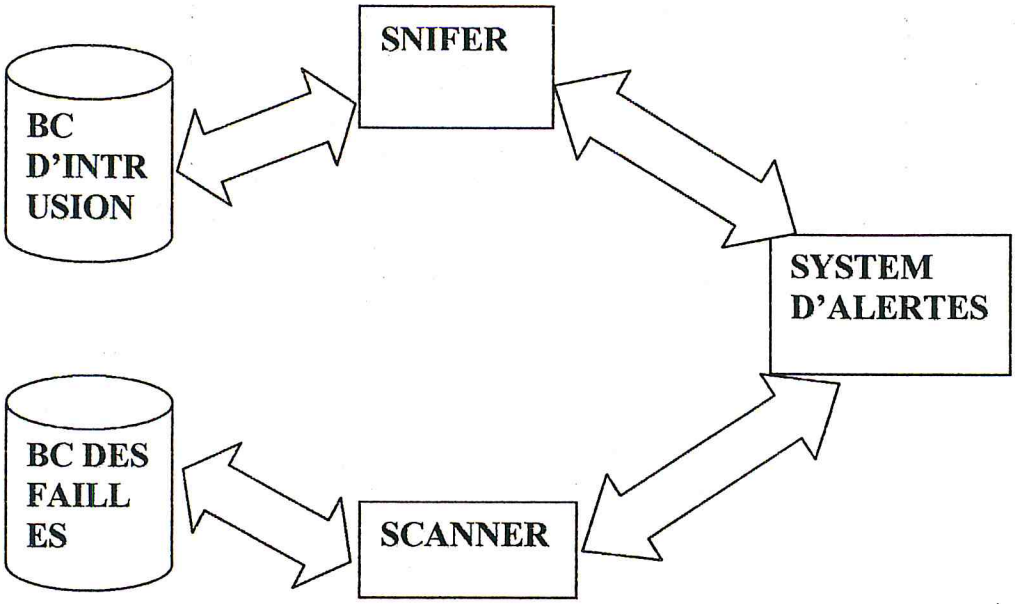


Figure 4.3 : les modules de notre scanner de sécurité

4.2OUTILS DE DEVELOPEMENT :

Pour implémenter notre system, on a opté pour l'environnement Windows (Windows NT4.0/2000/xp) qui est largement répandu dans les administrations et les différentes organisations. Dans ce qui suit nous argumenterons le choix du langage Visual C++ 7.0 (dot net) comme outils de développement ainsi que les APIs utilisées pour la capture et la construction des paquets.

4.2.1CHOIX DU LANGUAGE VISUAL C++ :

Le choix porté sur le langage MS Visual C++ 6.0 est dû, d'une part, au fait que la librairie utilisée pour la capture de paquets (*Libpcap*) est écrite en « C » ; d'une autre part, la DLL (Dinamyc Link Library) utilisée par Libpcap (*Packet.dll)* est non compatible avec les autres compilateurs comme celui de Borland (C++ Builder). Ajouter à cela les avantages offerts par Visual C++ 7.0 :

–C'est langage multi-thread : Un programme peut lancer et gérer l'exécution de plusieurs Threads en parallèle, ce qui offre la possibilité de profiter de la puissance offerte par les systèmes d'exploitation multi-thread comme Windows.

–La vitesse et la convivialité d'un environnement de développement visuel.

–La MFC (Microsoft fondation classes) : elle dispose de plusieurs structures de données munies de primitives facilitant la programmation réseau.

–Et bien sure c++ est un langage orienté objet qui a fait ces preuves depuis son apparition.

4.2.2 CAPTURE DE PAQUETS (SNIFER):

La capture de paquets du réseau est une opération de bas niveau qui permet d'intercepter les paquets directement de la carte réseau. Cela est nécessaire pour l'analyse du trafic réseau.

Les limitations des sockets (impossibilité de capturer tout le trafic réseau, dans le cas des sockets raw et nécessité de réservation de port pour le cas des sockets TCP/UDP) nous ont poussé à étudier d'autres outils qui les surpassent et qui consistent en l'utilisation de **Libpcap** qui interagit avec NDIS (Network Driver Interface Specification).

*** NDIS :**

NDIS - Network Driver Interface Specification - est une interface entre le système d'exploitation et la carte réseau qui permet la réception et l'envoi de paquets. Plus précisément NDIS est un ensemble de fonctions qui définissent la communication entre le(s) gestionnaire(s) de(s) carte(s) réseau et le(s) pilote(s) de protocoles (IP, IPX ...) [DEG 00].

La librairie NDIS (NDIS.sys) fournit une interface abstraite qui offre la possibilité de développer des pilotes pour la carte réseau. Les pilote de capture de paquets PACKET.SYS (pour Windows NT4.0/2000) et PACKET.VXD (pour Windows 9x/ME) sont développés en exploitent les fonctionnalités de NDIS pour la capture de paquets du réseau .Ils sont implémentés dans la structure de NDIS en tant que pilote de protocole parce qu'ils ont besoin de communiquer en même temps avec les pilotes réseau et avec les applications de niveau utilisateur [DEG 00].

***Libpcap :**

Libpcap (Packet Capture Library) est une API (Application Programing

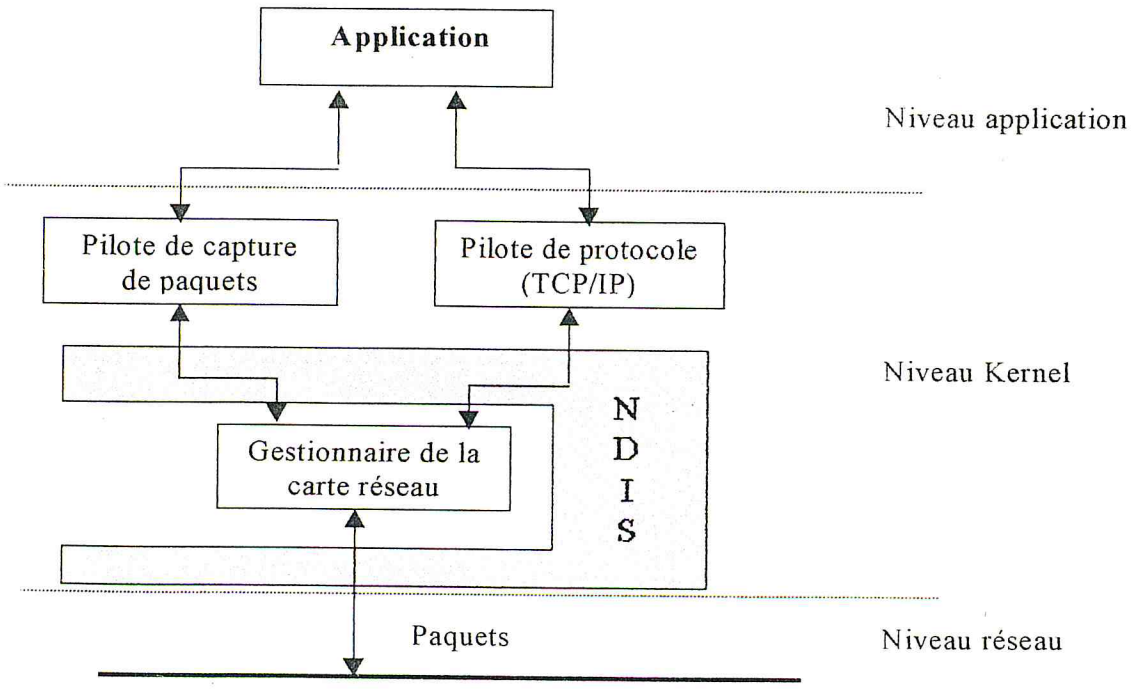


Figure 4.4 : Une simple structure de NDIS avec un pilote de capture de paquets.

Interface) qui offre une interface de haut niveau pour la capture de paquets. Elle a été développée initialement sous la plate-forme Unix (elle interagit avec BPF –Berkeley Packet Filter- qui est implémentée dans le noyau d'Unix), puis portée sur la plate-forme Win32 en exploitant les fonctionnalités de NDIS [DEG 00]. Les fonctions de Libpcap sont décrites dans [NET 01].

Parmis les fonctionnalités offertes par Libpcap on peut citer la capture du trafic réseau, le filtrage des paquets en exécutant le code des pseudo-machines BPF et la communication avec la carte réseau. Pour accomplir ces opérations Libpcap fait appel à PACKET.DLL (une librairie d'édition de liens dynamiques qui joue le rôle d'interface entre le pilote de capture de paquets PACKET.SYS ou PACKET.VXD et les applications de niveau utilisateur, cette DLL implémente une série de fonctions qui rendent simple la communication avec ce pilote) [DEG 00]. La figure 6.2 illustre la pile de capture de paquets dans notre système :

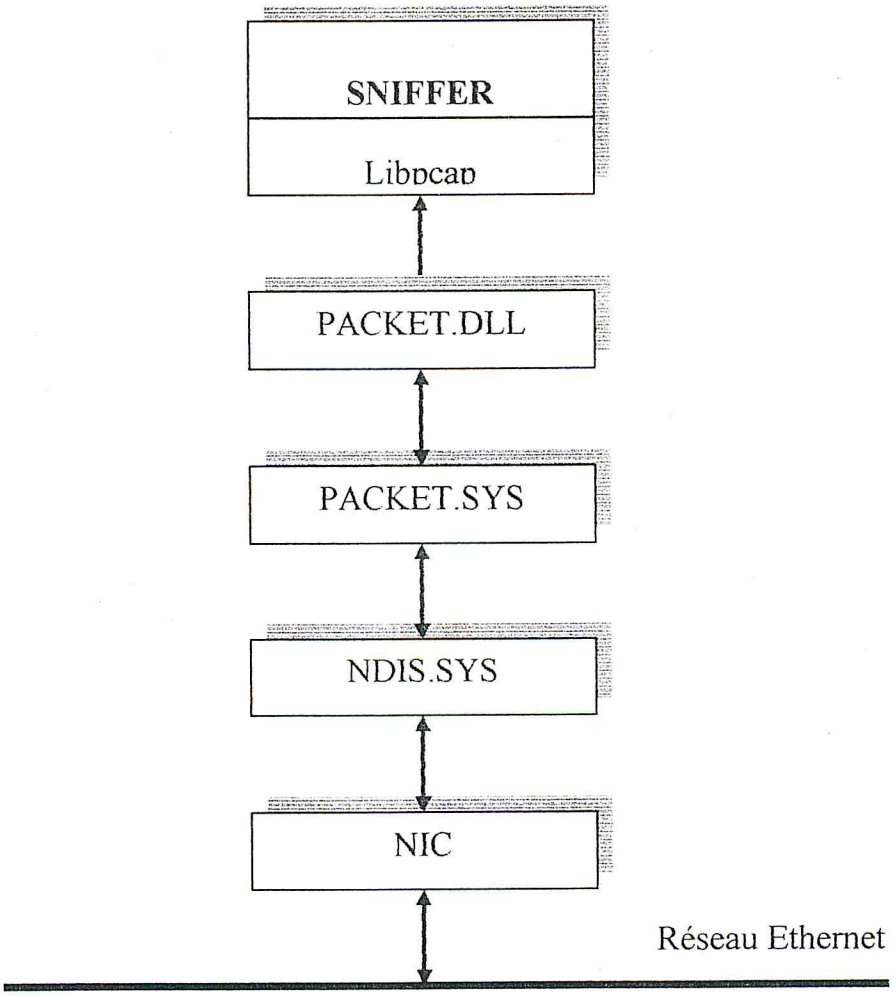


Figure 5.5: Communication entre Sniffer et la carte réseau pour la capture de paquets.

5. PERSPECTIVES

Le modèle que nous avons développé est nécessairement limité par le fait qu'il ne surveille que les attaques portant sur les couches Transport et IP. Or un grand nombre d'attaques existe pour la couche Applicative. Cependant, et de façon générale, la détection d'intrusion présente des limites et des problèmes et peut quelquefois se montrer impuissante dans certains cas :

Les coverts channels

La solution retenue ici ne permet pas de contrer les attaques par covert channels de type Loki, les logiciels qui permettent d'utiliser un canal a priori inoffensif pour faire transiter de l'info : par exemple loki fait communiquer le client et le serveur via des paquets icmp echo, AckCmd passe les firewalls en n'utilisant que des paquets ack. Ce genre de problème nécessite en fait de sniffer plus en profondeur, aucune solution n'existe contre ce danger potentiel, puisque même l'interception en profondeur peut être neutralisée par l'usage de la cryptographie pour camoufler les données en transit.

Les chevaux de Troie

Le même problème apparaît lorsqu'on cherche à se prémunir des trojan horses. Les trojans sont des applications s'apparentant aux virus dans la mesure où ceux ci sont installés souvent à l'insu de l'utilisateur. Une fois en place, toute personne sachant contacter le trojan peut prendre le contrôle de l'ordinateur infecté comme si il en était l'utilisateur actuel (certains logiciels poussent même le zèle jusqu'à permettre d'afficher l'écran de l'utilisateur piraté...). Une façon de détecter les trojans est donc de vérifier que les ports de communication habituellement ouverts par un trojan ne sont pas actifs sur la machine. Malheureusement, les trojans les plus récents sont hautement configurables, et peuvent donc être activés sur n'importe quel port. Ce genre de signature n'est donc plus efficace.

La seule méthode à peu près efficace pour détecter ce genre d'attaque repose sur le fait qu'un serveur doit être nécessairement installé sur la machine infectée pour que le covert channel ou le trojan soit mis en place : ainsi, une analyse rigoureuse des changements sur le disque dur d'une machine associée à l'observation éventuelle de trafic inexpliqué peut permettre de découvrir un covert channel ou un trojan en activité.

Enfin en peut ajouter les fonctions suivants pour améliore ce prototype :

- Une base de connaissance des intrusions qu'elle accepte la mise à jour de l'extérieur (administrateur) en XML par exemple ;
- Adapter un IDS active qui peut par exemple bloquer les ports automatiquement;
- Ajouter un module qui détecte les snifer installé dans le réseau ;
- Modifier le system d'alerte en intègre le wap (warless access protocole) pour alerter l'administrateur par **sms** vers son mobile.

6. CONCLUSION

Un des gros désavantages de la détection par signatures d'attaques est son manque de flexibilité et par conséquent sa vulnérabilité aux mutations : d'une part, il faut pour pouvoir définir une signature avoir déjà été confronté à l'attaque considérée. D'autre part, certaines de ces signatures se basent sur des caractéristiques volatiles d'un outil, comme par exemple le port qu'un certain trojan ouvre par défaut par exemple. Les caractéristiques retenues pour définir la signature sont donc fragiles, et les signatures extrêmement sensibles aux mutations. Un exemple actuel est l'outil ADMutate, qui permet de camoufler une attaque au niveau applicatif afin de la rendre non détectable par les systèmes de détection d'intrusion conventionnels utilisant des signatures. Contre ce genre de problème, une parade consiste à définir ce qu'est l'état de compromission, c'est à dire l'état attendu d'une machine pendant ou après une attaque. On peut alors essayer de détecter quand la machine entre dans cet état : on ne saura pas comment la machine a été attaquée si l'attaque était de type inconnu, mais on se sera quand même aperçu que quelque chose a eu lieu. Bien sûr, la difficulté majeure dans cette parade est de définir ce fameux état de compromission. On pourrait donc penser qu'une approche par apprentissage serait une bonne alternative. Cependant, il n'en est rien outre une convergence plutôt longue vers un modèle comportemental normal, rien n'empêche un pirate se sachant surveillé de rééduquer un tel système. Cependant l'approche qu'elle est actuellement en recherche c'est l'approche basée sur un system multi-agents.

CONCLUSION GÉNÉRALE

Il est clair que la sécurité est un des problèmes les plus sérieux que connaissent les entreprises qui ont des réseaux informatiques. Chaque entreprise doit donc savoir à quel point sa sécurité est nécessaire.
Quoi qu'il en soit, il faut retenir qu'un réseau totalement sécurisé est une utopie. Un réseau totalement sécurisé est un réseau fermé, auquel personne n'a accès, que se soit par voie informatique ou par voie physique.
Alors il ne sera jamais possible de sécuriser totalement un système d'information. Il y aura toujours des hackers de génie pour découvrir des nouvelles failles dans les systèmes. Mais on peut toujours rendre une intrusion plus difficile, et si le pirate n'a pas d'intérêt particulier à pénétrer cette entreprise plutôt qu'une autre, s'il éprouve trop de difficulté, il changera sûrement de cible, pour en trouver une plus facile.

De ce fait, nous avons réalisé cet outil « scanner de sécurité». L'outil réalisé est destiné à aider l'administrateur du réseau dans les phases de recherche et de détection des faiblesses de sécurité dans le réseau. Cet outil est destiné à faire la vérification ; et présente les méthodes de détection des différentes failles rencontrés utilisant différents protocoles grâce à sa portabilité due au langage de programmation « C++ » utilisé pour son développement.
Notre application visait à atteindre les buts suivants :

- **examiner le réseau,**
- **évaluation des risques,**
- **alerter l'administrateur de réseau.**

Le projet actuel, tel qu'il a été décrit, nous a permit de nous rendre compte, des problèmes qui se posent au niveau de la représentation des signatures, traduisant les attaques et ceux de l'intégration des capacités d'apprentissage dans les systèmes de sécurités.
Au final, on se rend bien compte que derrière le mot scanner de sécurité se cache plutôt un concept qu'un logiciel.

Ce projet nous a été d'un grand apport pédagogique, puisqu'il nous a permit de découvrir la sécurité informatique, qui est une branche de l'informatique qui nous était totalement inconnue, et de nous familiariser avec de nouvelles techniques de réalisation de logiciels autres que celles déjà vus durant nos études. Nous avons aussi enrichi nos connaissances sur les réseaux.

L'outil réalisé n'étant pas une panacée pour les problèmes des réseaux interconnectés, il en reste toujours des améliorations à apporter. Pour cela, le prototype réalisé est ouvert aux modifications et à l'enrichissement. Les modifications peuvent être apportées au niveau fonctionnel en améliorant et en enrichissant la base des signatures, et au niveau de l'implémentation en introduisant des capacités de détection automatique des failles.

BIBLIOGRAPHIE

[ALL 00] Julia Allen, Alan Christie, William Fithen, John Mackhugh, Jed Pickel , «State of Practice of Intrusion Detection Technologies. », Ed Stoner Carnegie Mellon University, http://www.sel.cmu.edu/publication /pubweb.html , 2000.

[BID 95] C.Bidan ET V.Issarny, « Un aperçu des problèmes de sécurité dans les systèmes informatiques », Institut de Recherche en Système Aléatoires –IRISA-, Octobre 1995.

[BEL 76] Bell (D. E.) ET LaPadula (L. J.), « Secure Computer Systems: Unified Exposition and Multics Implementation. », Rapport technique n° MTR-2997 Rev. 1, MITRE Corporation, Bedford, Mass, 1976.

[BIB 77] Biaba K. J, « Integrity Consideration for Secure Compuer System. », Rapport technique n° MTR-2997, MITRE Corporation, Bedford, Mass, 1977.

[BAC 00] Rebecca BACE, «An Introduction to Intrusion Detection & Assesment.», ICSA http://www.icsa.net/ , 2000 .

[CHA 02] Samuel Chabert « La sécurité » Université de Marne-la-Vallée Janvier 2002

[COZ 03] S. Coze et S. Heldebaume, « Les protocoles d'authentification», De l'IUP du Littoral sur le site www.guill.net

[CIS 98] CISCO SYSTEMS « Internetworking technology overview » http://cisco.com.

[DEG 00] Loris Degioanni, «Development of Architecture for Packet Capture and Network Traffic Analysis », Ecole Polytechnic de Torino (Italy) ,2000.

[DOS 83] Departement of Defense Standard. « Trusted Computer System Evaluation Criteria », *onrange book* , CSC-STD-001-83.

[DIF 76] W. Diffie et M. E. Hellman, « New Directions in Cryptography », *IEEE Transactions on Information Theory*, IT–22, no. 6:644–654, novembre 1976.

[DEB 98] H. Debar, M. Dacier & A.Wespi, « Towards a Taxonomy od Intrusion Detection Systems », Internal RZ 3030, IBM Zurich Research Laboratory, June 1998.

[FES 99] Olivier Festor, Nizar Ben Youssef, « Intégration du modèle de l'information de gestion CIM dans l'approche OSI», Institut Français de recherche en informatique, N° 3647, 7 avril 1999.

[FOR 97] S. Forrest, S. A. Hofmeyr, and A. Somayaji, « Computer immunology », Communication of of the ACM, 40(10):88-96, October 1997.

[GUI 00] Guillaume Desgeorge « La sécurité des réseaux » http://www.guill.net, 2000.

[GRE 03] PAUL Grégory
« MÉMOIRE TECHNIQUE ECRITURE D'UN FIREWALL EN JAVA »
Université de Paris, 30 mars 2003

[HPD 01] L.Henriet, P. Pezziardi et D. Schneider
« Le Livre Blanc de la Sécurité »
OCTO Technology, Septembre 2001.

[IRW 99] S. T. Irwin, Tom Bakey,
« SURVIVAL: Tips for staying alive in a competitive industry »,
Insurance Software Review, Fev/Mar 1999.

[ISO 89] Norme Internationale ISO 7498-2,
« Système de traitement de l'information Interconnexion des systèmes ouverts »,

[ISO 03] L'ISO 17799
http://www.iso-17799.com, 2003.

[ITS 91] Commission of the European Communities - ITSEC -
« Information Technology Security Evaluation Criteria »,
European Comunities, v1.2 édition, juin 1991.

[JEN 99] Christian Damsgaard JENSEN,
Thèse pour obtenir le grade de docteur de l'université Joseph Fourier,
Discipline : Informatique, 29 octobre 1999.

[LAR 96] Marc Laroche,
« Sécurité des Réseau : Analyse et Mise en Œuvre »,
Gouvernement du Canada, (CST) C.P. 9703, Terminus, Ottawa, Ontario, Janvier 1996.

[LAN 81] C. E. Landwehr,
« Formal Methods for Computer Security »,
ACM Computing Surveys, 13, no. 3:247–278, septembre 1981.

[LAI 92] X. Lai
« *On the Design and Security of Block Ciphers* »,
Hartung–Gorre, 1992.

[LAU 01] Maryline LAURENT-MAKNAVICIUS
« Sécurité d'Internet »
Technique de l'ingénieur, 2001

[MEB 99] B. Meeks, A. Boyle, et B. Sullivan,
« Hack attack knocks out FBI site »,
MSNBC, 26 mai 1999.

[MIC 01] Laboratoire Microsoft,
« Interconnexion en environnements hétérogènes avec Microsoft TCP/IP.»,
Supinfo, N° REF : 70-059-1.0, 2001.

[MON 02] André Mondoux
« Politique de sécurité »
Vol. 15, no. 6 / Direction Informatique, Juin 2002.

[NET 01] http://netgroup-serv.polito.it/winpcap/ 2001.

[NCS 87] National Computer Security Center,
« Trusted Network Interpretation of the TCSEC »,
Rapport technique, NCSC-TG-005, juillet 1987.

[NBS 77] N. B. of Standards,
« Data Encryption Standard »,
Number 46 in NBS, FIPS PUB, U. S. Department of Commerce, janvier 1977.

[NIC 99] M. Pascal Nicolas,
« Polyreseau »
U.F.R Sciences de l'Université d'Angers, 1999.

[NEE 78] R. M. Needham et M. D. Schroeder,
«Using Encryption for Authentication in Large Networks of Computers »
Communications of the ACM, 21, no. 12 :993–999, Décembre 1978.

[PIL 03] Jean-François Pillou,
« LES protocoles.»,
www.CommentCaMarche.net , GNU FDL, 2003.

[PRI 98] Jacques PRINTZ
« Génie logiciel »
Techniques d'ingénieur ; 1998.

[PUJ 97] M .Guy Pujolle
« L'architecture TCP/IP »,
Technique de l'ingénieur, H 2280, 1998.

[PUJ 98] M .Guy PUJOLLE
« Le modèle de référence OSI »,
Technique de l'ingénieur, H 1160, 1998.

[ROU 94] Serge Rouveyrol
« PROGRAMMATION RESEAU SUR TCP/IP L'INTERFACE DES SOCKETS »
E.N.S.I.M.A.G, Année Spéciale Informatique 1993/1994.

[RIV 78] R. L. Rivest, A. Shamir, ET L. Adleman,
«On a Method for Obtaining Digital Signatures and Public Key Cryptosystems»,

[RFC xx] Request for comments,
Les RFC peuvent être obtenue via le site www.ds.internic.net

[STO 89] C. Stoll,
« Le nid du coucou » Titre original « The Cuckoo's Egg »,
Albin–Michel, 1989.

[SEC 02] Securiteinfo
« Le grand livre de sécurité »
www.sécuritéinfo.com

[SYM 03] Symantec http://www.symantec.com/region/fr 2003.

[TAN 01] M. Androw TANENBAUM,
« Réseau »,
Prentice Hall, DUNOD no. 61250, janvier 2001.

[TEA 03] www.securite.teamlog.com

[TIL 99] David TILLOY, Support de cours
« Introduction aux réseaux TCP/IP.»,
Institut Universitaire de Technologie d'Amiens, 1999.

[OLO 92] Tomas Olovsson,
« A Structured Approach to Computer Security »
University of Technology Gothenburg SWEDEN, Technical Report n°122.1992.

[VAC 89] H. S. Vaccao and G.E. Liepin
« Detection of anomalous computer session activity »,
In proceedings of the IEEE Synposium on Security and Privacy. May 1989.

[VER 95] Floren Verriere & Nicolas Zuanon,
« La sécurité dans les réseaux »,
ENSIMAG 1995.

[WIA 03] Eric Wiatrowski
« La politique de certification : une stratégie business »
Ecole Supérieure d'Optique, 2003.
www.rd.francetelecom.fr/fr/conseil/mento20

1. Introduction :

Le titre de cette norme, "Code of practice for information security management", Il s'agit d'un recueil de bonnes pratiques dans la gestion de l'information, afin d'en assurer la sécurité. Les bonnes pratiques (*«Business Best Practices»*) permettent à l'entreprise de réagir efficacement et rapidement à toute circonstance sans devoir réinventer la roue.

ISO17799 est une démarche similaire dans le domaine de la sécurité informatique : une entreprise qui a la certification ISO17799 a été auditée et a été reconnue capable de faire du commerce électronique en toute sécurité avec ses partenaires commerciaux [WIA 03].

2. Historique simplifié de la norme ISO 17799

- 1995 : Parution de la norme UK BS 7799 ;
- 1998 : Mise en place du schéma de certification UK. Notoriété et succès international de la norme ;
- 1/12/2000 : Adoption de BS 7799-1 en tant que norme internationale par l'ISO : ISO 177799 ;
- 2002 : Processus de révision lancé [ISO 03].

3. Les dix chapitres de la norme ISO 17799

Une trousse globale Le principal attrait de l'ISO 17799 est qu'elle offre une trousse complète de procédures axées sur la mise en place et la gestion des politiques de sécurité à l'intérieur des entreprises. Alors qu'on avait reproché à la norme BS 7799 d'être trop rigide, ISO 17799 est suffisamment «vague» pour s'adapter à la plupart des contextes culturels et informatiques.

Cette norme ne contient pas moins de 128 points répartis dans 10 chapitres avec une cohérence certaine. La plupart des actions logiques de sécurité y sont traitées :

- Contrôle d'accès ;
- Transport du courrier physique ;
- Plan de secours ;
- Maîtrise de la sous-traitance…

3.1 Politique de sécurité

Ce chapitre mentionne notamment la nécessité pour l'entreprise de disposer d'une politique de sécurité et d'un processus de validation et de révision de cette politique.

L'existence même de cette mesure (non technique !) montre que l'ISO 17799B est avant tout un immense catalogue de "bonne pratique" pour gérer de manière sécurisée ses informations [MON 02].

3.2 Organisation de la sécurité

Ce chapitre comporte 3 parties :

A. Une partie traite de la nécessite de disposer au sein de l'entreprise d'une organisation dédiée à la mise en place et au contrôle des mesures de sécurité en insistant sur :
   - l'implication de la hiérarchie et sur la coopération qui devrait exister entre les différentes entités de l'entreprise ;
   - la désignation de propriétaires de l'information, qui seront responsables de leur classification ;
   - l'existence d'un processus pour la mise en place de tout nouveau moyen de traitement de l'information ;

B. Une deuxième partie traite des accès aux informations de l'entreprise par une tierce partie. Ces accès doivent être encadrés par un contrat qui stipule les conditions d'accès et les recours en cas de problèmes ;

C. Une troisième partie indique comment traiter du cas où la gestion de la sécurité est externalisée (outsourcing) [MON 02].

3.3 Classification des informations

Ce chapitre traite de la nécessité de répertorier l'ensemble des informations (ou types d'information) de l'entreprise et de déterminer leur classification. La mise en place d'une classification de l'information doit s'accompagner de la rédaction de guides pour la définition des procédures de traitement de chaque niveau de classification.
On peut noter en effet qu'il n'est pas suffisant de donner des niveaux de classification, encore faut-il donner les moyens aux personnels pour mettre en oeuvre les procédures de traitement [MON 02].

3.4 Sécurité du personnel

Ce chapitre mentionne trois types de mesures :

- lors du recrutement de personnel, il est tout aussi important d'enquêter sur le niveau de confiance que l'on peut accorder aux personnes qui auront accès à des informations sensibles que de mentionner dans les contrats d'embauche des clauses spécifiques à la sécurité comme une clause de confidentialité ;
- une sensibilisation à la sécurité doit être proposée à toute personne accédant à des informations sensibles (nouvel arrivant, tierce partie) ;
- l'ensemble du personnel doit être informé de l'existence et du mode d'emploi d'un processus de remontée d'incidents [MON 02].

3.5 Sécurité de l'environnement et des biens physiques

Ce chapitre traite de toutes les mesures classiques pour protéger les bâtiments et les équipements :

- délimitation de zone de sécurité pour l'accès aux bâtiments (attention aux accès par les livreurs)
- mise en place de sécurité physique comme la lutte contre l'incendie ou le dégât des eaux
- mise en place de locaux de sécurité avec contrôle d'accès et alarmes, notamment pour les salles machines
- mise en place de procédures de contrôle pour limiter les vols ou les compromissions
- mise en place de procédures pour la gestion des documents dans les bureaux [MON 02].

3.6 Administration

Ce chapitre traite des points suivants :

- rédiger et mettre à jour l'ensemble des procédures d'exploitation de l'entreprise (que ce soit pour de l'exploitation réseau, système ou sécurité) ;
- rédiger et mettre à jour les critères d'acceptation de tout nouveau système ;
- prévoir un planning pour l'achat de composants ou matériels pour éviter tout interruption de service ;
- mettre en place un certain nombre de politique organisationnelle et technique (anti-virus, messagerie, diffusion de document électronique en interne ou vers l'extérieur, sauvegarde et restauration, etc.) [MON 02].

3.7 Contrôle d'accès

Ce chapitre comprend beaucoup de propositions de mesures par rapport aux autres chapitres. Sans être exhaustif, on peut cependant retenir :

- la nécessité pour l'entreprise de disposer d'une politique de contrôle d'accès (qui a droit à quoi et comment il peut y accéder)

- la mise en place d'une gestion des utilisateurs et de leurs droits d'accès sans oublier la révision de ces droits (gestion de droits, gestion de mot de passe ou plus généralement d'authentifiant)
- la responsabilité des utilisateurs face à l'accès aux informations (ne pas divulguer son mot de passe, verrouiller son écran quand on est absent par exemple)
- des propositions de mesures pour mettre en oeuvre la politique de contrôle d'accès comme l'utilisation de la compartimentation de réseaux, de firewalls, de proxies, ..., la limitation horaire d'accès, un nombre d'accès simultanés limité, etc.
- la mise en place d'un système de contrôle de la sécurité et de tableaux de bord
- l'existence et la mise en place de procédures concernant le télétravail

Ce chapitre est très représentatif de la traduction française du titre de la norme "codes de bonnes pratiques" à savoir qu'il donne les mesures à mettre en place pour gérer ses informations de manière sécurisée mais il propose aussi des exemples de mise en oeuvre comme l'utilisation d'un firewall [MON 02].

3.8 Développement et maintenance

Ce chapitre, de la même manière que précédemment, propose des mesures incontournables comme des exemples de mise en oeuvre. Sans être exhaustif, on peut retenir :

- la nécessité d'intégrer les besoins de sécurité dans les spécifications fonctionnelles d'un système
- des conseils de développement comme la mise en place d'un contrôle systématique des entrées sorties au sein d'un programme
- des propositions d'intégration de services de sécurité comme le chiffrement, la signature électronique, la non répudiation, ce qui nécessiterait pour l'entreprise la définition d'une politique d'usage et de contrôle d'outils à base de cryptographie ainsi qu'une politique de gestion des clés associées
- la mise en place de procédures pour l'intégration de nouveaux logiciels dans un système déjà opérationnel
- la mise en place d'une gestion de configuration [MON 02].

3.9 Plan de continuité

Ce chapitre traite de la nécessité pour l'entreprise de disposer de plans de continuité ainsi que de tout le processus de rédaction, de tests réguliers et de mise à jour de ces plans [MON 02].

3.10 Conformité légale et audit de contrôle

Ce chapitre traite pour l'essentiel de deux points :

- la nécessité pour l'entreprise de disposer de l'ensemble des lois et réglements qui s'appliquent aux informations qu'elle manipule et des procédures associées
- la mise en place de procédures pour le déroulement d'audit de contrôle [MON 02].

4. Le passage entre ISO 17799 et BS 7799

Une entreprise allant vers la certification BS 7799 bénéficiera d'un avantage concurrentiel certain vis-à-vis de ses compétiteurs non certifiés. En effet, quand le choix se propose, les clients ont une tendance naturelle à se diriger vers les solutions certifiées... Ainsi, la certification garantit la mise en place de six mesures complémentaires :

- Renforcer la sécurité de l'entreprise ;
- Rendre effective la gestion de la sécurité ;
- Sécuriser les partenariats et l'e-commerce ;
- Augmenter la confiance des clients ;

- Fiabiliser et préciser les audits de la sécurité ;
- Réduire les failles du système [ISO 03].

5. Le statut évolutif de la norme ISO 17799

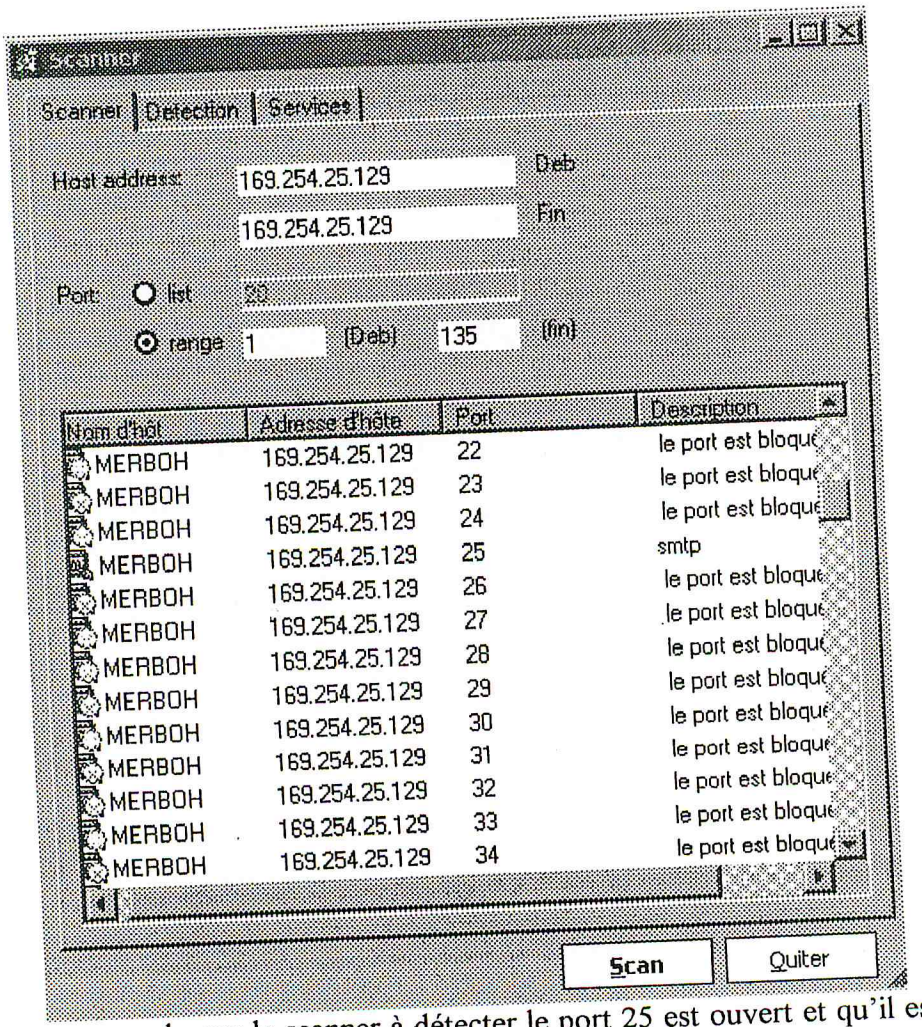
Pour rester en phase avec son environnement, une norme doit pouvoir évoluer avec le temps. Par conséquent, l'organisation ISO travaille actuellement sur une nouvelle version pour répondre aux nouvelles attentes de son audience sans cesse croissante. Ses extensions futures seront développées à partir de la base existante [SYM 03].

1. Introduction

L'application réaliser est programme MFC (un ensemble de classes C++ qui encapsulent plusieurs functions des applications implémenté pour le système d'exploitaion Microsoft Windows) comporte trios principaux modules, un scanner de ports, un système de détection (snifer) et un service de mise à jour la base des attaques.
Le programme à était compiler à l'aide de Microsoft Visuel C.NET. On a utilise pour l'interface la bibliothèque ProfUIS221, Winsock API pour le premier module, la bibliothèque Libcap pour le module snifer.
Ce prototype peut exécuter ce trois modules en même temps ce qui permit le de savoir quel sont les ports ouverts en même moment la détection des attaque peut être lancer dans l'arrière plan.

2. Le module scanner

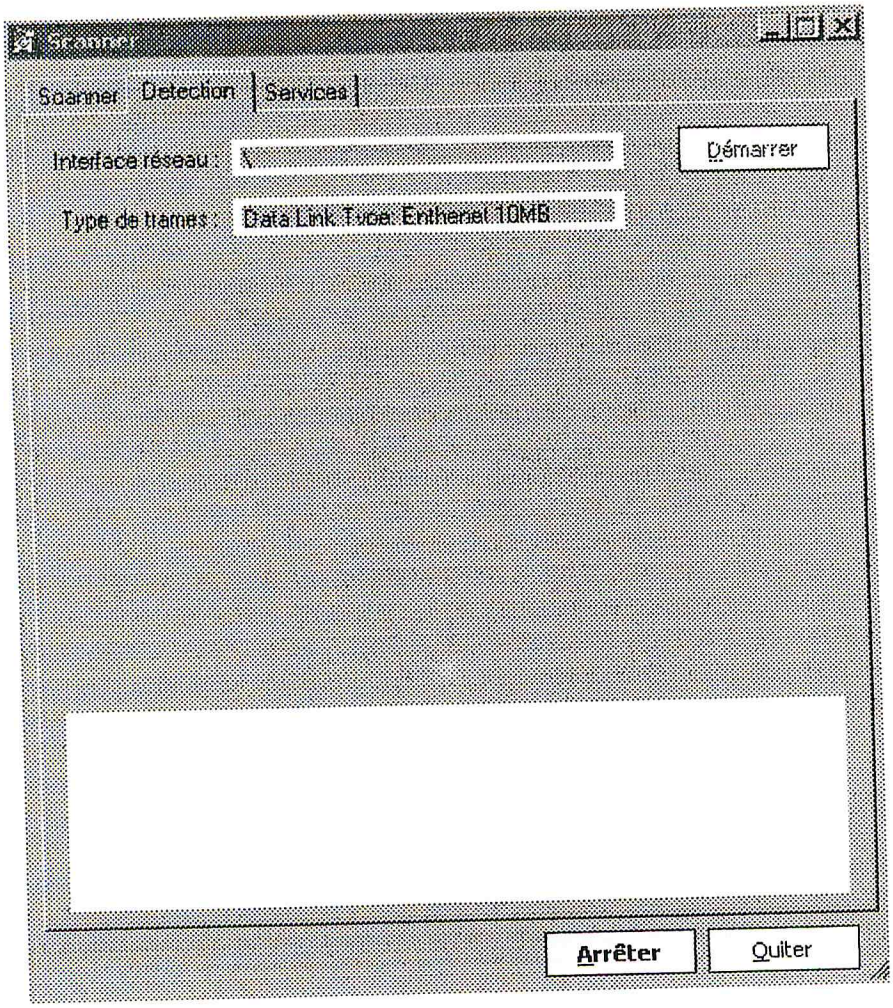
C'est un simple scanner de port mais effective, il utilise sa propre classe de sockets (pas la classe `CSocket` qui marche avec MFC) et un petit squelette base sur les classes `CPropertySheet/CPropertyPage`.
Le principe c'est d'envoyer une socket pour chaque port, si la socket est valide alors le port concerné est ouvert. La Winsock API peut être utiliser en trois modes, blocking, non-blocking and asyncronous mode. Notre scanner utilise la troisième (asyncronous) mode (un port à la fois sans ploquer l'UI).
On peut scanner un seul port ou plusieurs ports comme l'indique la figure suivante :



On voit dans ce exemple que le scanner à détecter le port 25 est ouvert et qu'il est utiliser par le protocole SMTP, et les autres sont bloquer.

3. le module détection

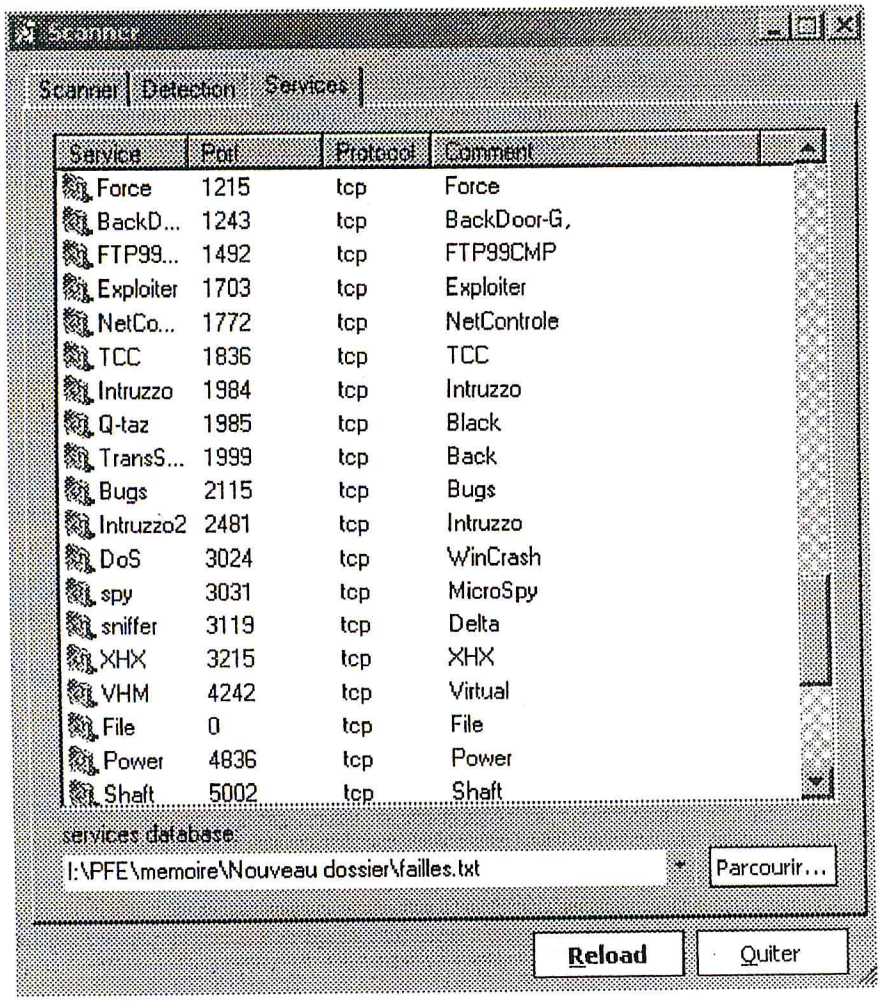
Ce module est le plus importants car il à comme peut de capter les trames qui circulent dans le réseaux et les analyser afin de détecter une attaque à l'aide des classes de la méthode pattern macthing ; il détecte aussi la carte réseau et permit de visualiser les entêtes des paqeuts,



Dans ce cas rien ne circule dans le réseau.

4. Le module service

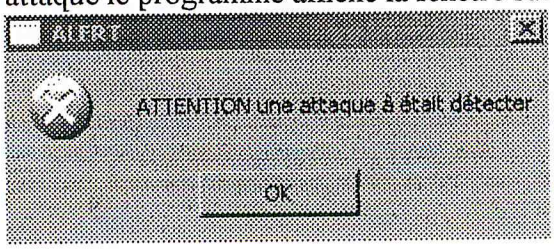
Ce module permit la détection des attaques de type cheveux de trois en comparent les ports ouvert avec les ports utilisé par ces ver. Ces ports sont définit dans une base de données et peut être mise a jour en rechargent une nouvel base de données.



5. L'alert

En cas de détection d'une attaque le système envoi un message SNMP au post serveur contenant le nom de post attaqué et le type d'attaque.

Sur le post concerné par l'attaque le programme affiche la fenêtre suivants



6. Plateformes supportées :

Processeurs : Intel

Systèmes d'exploitation: Windows 98- Windows NT4 SP4 - Windows 2000- Windows XP

7. Informations complémentaires :

L'installation de scanneur nécessite:

32 MB de mémoire

4 MB d'espace disque libre