# Attaques des réseaux

*par* **Laurent LEVIER**
*Certified Information Systems Security Professional (CISSP)*
*Certified Information Security Manager (CISM)*
*Officier de sécurité du réseau interne, Equant Télécommunications*



*Depuis les années 1980 est apparu le concept de réseau informatique. Initialement, il s'agissait de réseaux locaux (LAN) à 10 Mbits/s, ce qui représentait le haut débit pour cette époque, puis la vitesse a évolué pour dépasser le gigabit par seconde. Les premiers protocoles alors utilisés sur ces réseaux souvent Ethernet et Token Ring étaient non routables tels que Netbeui ou IPX. Puis ces réseaux ont évolué vers des tailles plus conséquentes (MAN ou WAN), en persistant souvent à utiliser des protocoles non routables ou en utilisant les premiers protocoles bureautiques routables comme IPX/SPX.*

*C'était le temps où la sécurité des réseaux informatiques n'était pas une véritable préoccupation pour tout un chacun car le piratage sévissait principalement sur les réseaux téléphoniques (« phreaking »).*

*Avec l'avènement du réseau des réseaux Internet est arrivée la vulgarisation de l'informatique vers le grand public. Il devint alors possible pour tous d'accéder à Internet et à la masse d'informations qu'il contient. Ce fut aussi l'apparition du protocole de routage qui est maintenant le plus utilisé tant sur Internet qu'au sein des réseaux privés d'entreprises ou de particuliers : TCP/IP. Suite à l'ouverture d'Internet au monde, les entreprises ont été contraintes de se connecter à ce réseau pour profiter de cette vitrine mondiale. Malheureusement, si Internet a apporté une formidable révolution de l'informatique et de la circulation mondiale de l'information, il a également mis à la disposition de personnes mal intentionnées de nouveaux moyens d'accéder illégalement à des données privées, qu'elles soient au sein de l'entreprise ou chez un particulier, et ceci avec un risque bien moindre puisque sans intrusion physique.*

*En effet, il est nécessaire pour l'entreprise d'être connectée à Internet afin d'exploiter ses mines d'informations. Par conséquent, il devient possible pour*

*quiconque sur Internet d'accéder aux ressources de l'entreprise si elle n'a pas mis en place de protections appropriées.*

*Par ailleurs, d'autres technologies de réseau telles que le réseau sans fil (« wireless ») sont nées. Au premier abord, cela semble une avancée pour les utilisateurs qui peuvent enfin s'affranchir de la contrainte de connexion filaire. Mais si on se focalise sur la sécurité de cette évolution, elle constitue en fait une régression si elle n'est pas utilisée intelligemment, car il devient possible d'accéder à un réseau privé sans lui être physiquement connecté.*

*Enfin, toutes ces technologies contiennent toujours plus ou moins des erreurs de conception ou de configuration. Ces erreurs sont la plupart du temps publiées sur Internet avant qu'elles ne soient corrigées, ce qui permet à des personnes mal intentionnées de les exploiter dans le but de pénétrer les réseaux privés reliés à Internet.*

*Ces personnes, que nous qualifierons simplement d'intrus ou de pirates, ont alors besoin de mieux connaître le réseau qu'elles comptent attaquer et franchissent pour cela une série d'étapes mettant en œuvre différentes techniques selon le profil de leur victime.*

*Leur but est de cartographier le réseau afin d'en repérer les faiblesses et les cibles les plus intéressantes. Mais il peut être aussi de récupérer des informations circulant sur le réseau telles que les données d'authentification ou les caractéristiques d'un contrôle d'accès afin de les exploiter pour passer outre un autre mécanisme de sécurité. Cela peut être également la modification du comportement du réseau afin de se placer dans une position permettant de mener à bien telle ou telle attaque, afin de gagner des privilèges. Cela peut enfin être simplement une volonté de rendre le réseau inopérant, que ce soit par simple désir de nuire comme le font les vers informatiques (« worms ») ou par dépit face à un réseau particulièrement résistant à l'intrusion. Nous allons présenter les techniques et méthodes que doivent utiliser ces pirates afin d'atteindre ces buts.*

*Enfin, il est important de noter que ce document présente des principes, techniques et méthodes d'attaques qui ne sont souvent possibles que lorsque l'équipement du réseau n'est pas correctement sécurisé. Le risque et la possibilité d'une attaque peuvent être considérablement réduits si les équipements sont bien configurés, si des mots de passe efficaces sont utilisés, si les bogues, quels qu'ils soient, sont corrigés et les mises à jour appliquées. Malheureusement, ces attaques restent trop souvent réalisables car les équipements des réseaux sont la plupart du temps insécurisés soit parce que l'administrateur du réseau n'est pas suffisamment compétent, soit parce qu'il n'y a pas de culture de sécurité dans l'entreprise ou au sein de l'équipe d'administration du réseau.*

Un tableau récapitulatif des sigles et abréviations peut être consulté dans la partie « Pour en savoir plus » [Doc. H 5830].

# 1. Cartographier le réseau

En premier lieu, l'intrus doit se faire une idée de ce qu'il s'apprête à attaquer afin de sélectionner ses cibles. De son point de vue, le réseau qu'il veut attaquer n'est qu'un sous-réseau (*subnet*) dont seulement un certain nombre d'adresses sont utilisées. Si l'intrus peut exploiter des informations fournies par des services publics tels que le service de noms de domaine (DNS) pour obtenir quelques adresses assurément utilisées, il ne peut en revanche avoir aucune certitude sur les autres adresses et doit donc pour cela trouver l'information par lui-même. Il peut alors avoir recours au balayage (*scanning*) du réseau cible.

## 1.1 Balayage du réseau

Il existe différentes techniques pour balayer un réseau cible. Celles-ci sont bien sûr plus ou moins discrètes et donc détectables par d'éventuels mécanismes de protection.

### 1.1.1 Balayage classique *via* ICMP : `ping`

La méthode la plus classique repose sur l'exploitation d'un service disponible depuis Internet et ceci même si des mécanismes de protection du réseau cible existent. Ce service est souvent connu sous le nom d'une commande : `ping`.

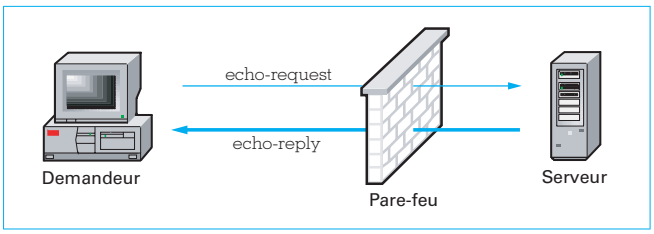


**Figure 1 –** ping

La fonction de ping est d'offrir à une personne sur Internet un mécanisme permettant de déterminer si une machine est bien active. Il ne s'agit pas de dire si les services du réseau HTTP ou FTP par exemple sont disponibles, mais simplement si la fonction réseau du système d'exploitation tournant sur une adresse IP est opérationnelle. Le principe est très simple. La machine désireuse de savoir si une autre est active envoie un paquet ICMP echo-request et reçoit en retour, si la machine réceptrice est bien active, un paquet ICMP echo-reply. Comme c'est fréquemment le cas, un **pare-feu** peut être sur le chemin de la demande de ping et laisser passer ce type de trafic (figure **1**).

Dans le cas du balayage d'un réseau par ping, l'intrus utilise simplement un programme qui vérifie les adresses du sous-réseau et note celles qui répondent.

Cette méthode, en général efficace pour une analyse discrète, puisqu'elle ne s'appuie sur aucun comportement offensif, n'est cependant pas indétectable. Il est en effet fréquent qu'un pare-feu stocke les traces (*logs*) des paquets qu'il a acheminés d'une interface à l'autre. L'analyse de ces traces permet donc de détecter qu'une même machine est en train de tester chaque adresse du réseau.

Il devient alors utile d'utiliser une variante plus discrète, puisqu'elle ne nécessite l'envoi que d'un seul paquet pour balayer la totalité du réseau cible. Il s'agit d'utiliser les *broadcasts*.

Le *broadcast* est une adresse IP qui correspond à la plus haute adresse disponible dans un sous-réseau. Cette adresse est utilisée pour envoyer un paquet réseau à toutes les machines présentes dans le sous-réseau correspondant, car dans la définition du protocole TCP/IP, il est stipulé qu'une machine qui reçoit une demande vers son adresse de *broadcast* doit y répondre.

**Exemple :** pour un sous-réseau 192.168.0.0/24 (une classe C), les adresses possibles sont comprises entre 192.168.0.0 et 192.168.0.255. Cette dernière correspond au *broadcast* de ce sous-réseau particulier.

Le principe du balayage par *broadcast* est donc des plus simples. Il suffit d'envoyer la demande ICMP echo-reply (étape 1 de la figure **2**) vers l'adresse de *broadcast* du réseau cible et toutes les machines présentes y répondent (étape 2 de la figure **2**).

Il est bon de noter que si un pare-feu est présent sur le chemin entre le demandeur et les serveurs, il n'a pas de raison de bloquer les ICMP echo-reply en provenance de chaque serveur puisqu'il s'agit d'une réponse normale selon son algorithme d'analyse.

Nous verrons également (§ 3.3) que le *broadcast* est très souvent utilisé pour les attaques en déni de service puisqu'il permet d'amplifier le nombre de paquets et donc de saturer la bande passante du réseau cible. Il est donc recommandé de bloquer systématiquement les adresses de *broadcast* en entrée sur un pare-feu pour se protéger des réseaux extérieurs, mais aussi en sortie si l'on suspecte son propre réseau d'être à l'origine de l'envoi d'un nombre inconsidéré de *broadcasts*.

### 1.1.2 Fonction traceroute

Tout utilisateur a eu besoin au moins une fois de la fonction traceroute. Présente sous ce nom sous Unix ou sous le nom tracert

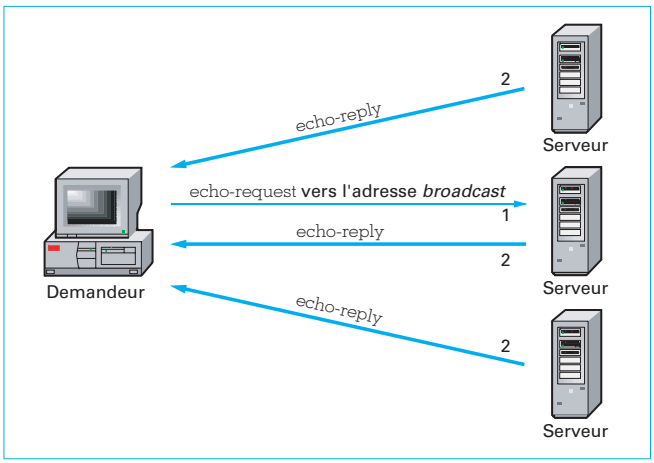


**Figure 2 –** ping **en *broadcast***

sous Windows, elle permet de déterminer le chemin réseau entre deux équipements.

Mais en réfléchissant bien, que fait la fonction traceroute ? Elle permet d'établir la topologie d'un réseau au-delà des systèmes qui le composent. Avec les techniques de balayage que nous avons présentées, l'intrus a pu dresser une liste des machines présentes sur un sous-réseau. Avec traceroute, il peut obtenir des chemins vers d'autres réseaux et ainsi commencer à construire une véritable topologie du réseau cible avec ses points d'interconnexion, leurs éventuels mécanismes de contrôle d'accès, etc.

Pour fonctionner, traceroute envoie un paquet avec la durée de vie (TTL) égale à 1. Le premier routeur se fait alors connaître en répondant par un message d'erreur ICMP indiquant que le paquet ne peut être acheminé parce que le TTL a expiré.

Traceroute renvoie alors un paquet avec le TTL égal à 2. Dans ce cas, le second routeur sur le chemin répond avec le TTL expiré comme avait répondu le premier routeur quand le TTL était égal à 1. Traceroute recommence jusqu'à ce que l'adresse de destination soit atteinte. La figure **3** présente ce principe. Le résultat final fournit le chemin emprunté entre l'adresse IP source et l'adresse IP de destination.

### 1.1.3 Découvrir les routeurs

Il existe également des techniques pour découvrir particulièrement les équipements assurant des fonctions de routage. Ainsi, écouter le réseau permet de voir passer les paquets *multicasts* en provenance des routeurs comme les trames Hello qui permettent aux routeurs de se présenter entre eux ou les mises à jour d'un protocole de routage.

Mais il est également possible de faire des requêtes auxquelles les routeurs répondront. Ces requêtes peuvent s'appuyer sur une demande ICMP de découverte de routeur (*ICMP router discovery*) ou des requêtes de protocoles de routage (OSPF, BGP...).

**Exemple :** un intrus peut envoyer des requêtes IRDP (ICMP Router Discovery Protocol), également appelées sollicitations de routeur (*router solicitations*), vers l'adresse de *broadcast* afin de connaître la route par défaut.

### 1.1.4 Balayage SYN

Les techniques que nous venons d'aborder fonctionnent en général assez bien, mais présentent l'inconvénient majeur d'être tracées (*logs*) par les équipements de sécurité sur le chemin réseau entre l'intrus et sa victime. Il existe une technique qui, à

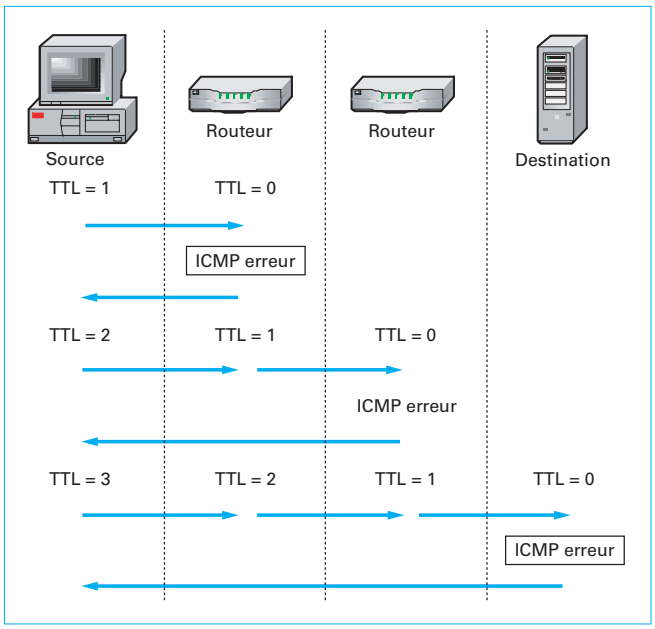


**Figure 3 – Fonctionnement de** traceroute

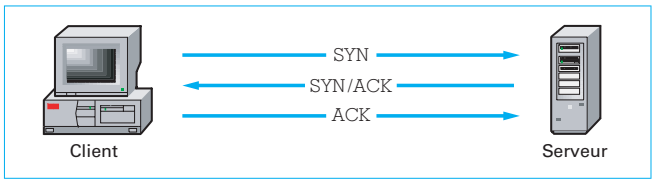


**Figure 4 – Établissement d'une session TCP**

l'inverse, présente l'avantage de ne pas être toujours tracée : le balayage SYN.

Afin de bien comprendre le principe de cette technique qui est surtout utilisée dans le balayage de ports, il est important de rappeler comment s'établit au départ une session TCP entre un client et un serveur. Comme le montre la figure **4**, le client envoie un paquet SYN vers le serveur qui lui répond (si le port est en écoute) avec un paquet SYN/ACK. Enfin, le client confirme la réception du SYN/ACK avec un ACK et vient ensuite une longue suite d'échanges où l'émetteur envoie ses données et reçoit en retour un paquet ACK, jusqu'à l'initiation de la fin de l'échange.

Dans la technique du balayage SYN, le client envoie un paquet SYN comme dans toute demande de session TCP et reçoit donc une réponse. Si le port est en écoute, un SYN/ACK est retourné, sinon c'est un RST. En l'absence de réponse, l'adresse IP est considérée inactive. Une fois la réponse reçue, le client note l'état de la machine (selon la réponse) mais ne continue pas la phase d'initiation de la session TCP.

Ainsi, le pare-feu éventuel ne voit pas une session TCP complète et considère qu'il n'y a pas eu d'échange d'information. Il ne trace donc pas l'échange alors que, pourtant, un intrus a bien pu détecter l'état d'une adresse IP (en fait d'un port particulier de l'adresse IP cible).

Il existe des variantes jouant sur le paquet mais il s'agit là davantage de techniques de balayage de ports que d'adresses IP, donc d'une attaque visant à pénétrer un système plutôt que de détecter sa présence. Nous les aborderons donc plus finement dans [H 5 832], qui traite des techniques d'attaque contre un équipement dans le but d'une intrusion.

### 1.1.5 Empreinte du réseau d'un système d'exploitation

Bien que cette technique soit orientée vers le balayage de ports plutôt que d'adresses IP, nous devons faire un aparté sur les mécanismes permettant de déterminer le type de système d'exploitation derrière une adresse IP.

Normalement, le seul moyen de déterminer quel est le système d'exploitation d'un équipement branché sur un réseau est de récupérer les bannières des services qu'il offre. Par exemple, en se connectant sur le serveur HTTP (port 80) de Microsoft comme le montre l'encadré **1**, on peut déterminer quel logiciel utilise Microsoft pour héberger son site.

Cette technique présente l'inconvénient d'être détectable par le serveur (qui trace la transaction), tout en n'étant pas toujours possible car certains services réseaux particulièrement intéressants (telnet, ssh, ftp...) peuvent être filtrés par un pare-feu. Mais une nouvelle manière de procéder, qui n'a pas besoin d'application particulière, permet d'obtenir une réponse souvent plus précise que l'ancienne méthode.

Cette technique, appelée *TCP fingerprinting*, est basée sur le comportement réseau unique de chaque système d'exploitation et n'a en fait besoin que de deux ports réseaux sur la machine cible pour pouvoir fonctionner : un port en écoute et un port qui n'écoute pas.

Le logiciel client chargé de déterminer l'empreinte envoie des séries de paquets en jouant sur les bits de ceux-ci (FIN, URG, PSH...), mais aussi la taille, les options possibles et leur valeur, et note la réaction de la couche réseau du système cible.

**Exemple :** l'envoi d'un paquet FIN (habituellement utilisé pour terminer une session) vers un port en écoute ne devrait pas engendrer de réponse, mais Windows ou les équipements Cisco répondent par un paquet RST.

**Encadré 1 – Bannière du site HTTP de Microsoft**

```
$ telnet www.microsoft.com 80

Trying 207.46.156.188...
Connected to www2.microsoft.akadns.net.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Connection: close
Date: Fri, 21 May 2004 06:53:14 GMT
Server: Microsoft-IIS/6.0
Content-Length: 38913
Content-Type: text/html
```

**Encadré 2 – Empreinte réseau d'une machine sur Internet par nmap**

```
Interesting ports on address:
PORT STATE SERVICE
21/tcp open ftp
23/tcp closed telnet
Device type: general purpose
Running: FreeBSD 4.X
OS details: FreeBSD 4.7 - 4.8-RELEASE
Uptime 21.851 days (since Thu Apr 29 12:41:59 2004)
1 IP address (1 host up) scanned in 7.203 seconds
```

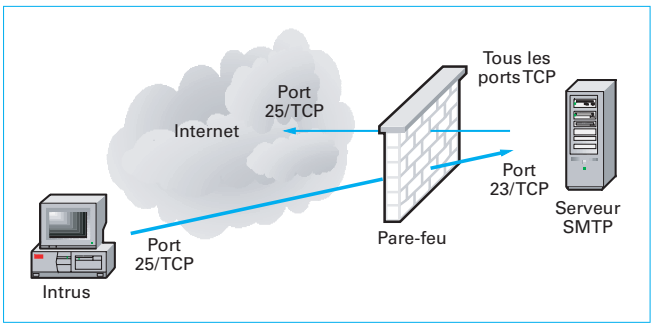


**Figure 5 – Traverser un pare-feu en fixant le port source**

Par recoupements, le logiciel construit des hypothèses et les affine jusqu'à finir la plupart du temps par sélectionner un système d'exploitation précis avec son numéro de version. Ainsi, en ne ciblant que les ports 21/TCP (FTP, ouvert) et 23/TCP (telnet, fermé) d'une machine, le logiciel nmap fournit en résultat l'encadré **2**.

Ainsi, la précision est assez grande et des renseignements supplémentaires tels que la durée de fonctionnement du système d'exploitation sont même fournis.

## 1.2 Traversée du filtrage d'un pare-feu

Dans la plupart des cas, les réseaux des entreprises sont protégés par un pare-feu, ou *firewall* [TE 7 550]. Celui-ci peut être un véritable pare-feu, mais aussi un simple routeur configuré de manière filtrante. Une des missions du pare-feu est de garder des traces (*logs*) des flux réseaux qu'il a laissé passer ou qu'il a bloqués, mais sa mission principale reste le contrôle d'accès et donc de laisser passer uniquement les flux autorisés par l'administrateur de la politique de sécurité.

Malheureusement, il existe des techniques pour passer outre ces contrôles d'accès par l'exploitation de faiblesses du logiciel du pare-feu ou d'erreurs de configuration de la politique, souvent dues à l'inexpérience de l'administrateur.

■ **Fixer le port source :** dans le cas où le pare-feu n'est qu'un simple routeur utilisant des listes de contrôle d'accès (ACL) ou un pare-feu qui ne saurait détecter qu'un flux correspond au trafic retour d'une session sortante déjà initiée, il est possible de passer outre les règles de filtrage appliquées par cet équipement sur le réseau en modifiant simplement le port source du paquet émis (*source porting*).

Comme le montre la figure **5**, notre routeur, ou pare-feu, a donc pour mission d'autoriser par exemple les flux partant des ports TCP du serveur situé sur le réseau de l'entreprise, à condition que ces flux visent n'importe quelle machine sur Internet sur le port 25/TCP. Aucune condition n'est émise sur le numéro de port source. Il s'agit là d'une règle typique pour du trafic SMTP permettant aux serveurs de messagerie de s'envoyer des messages électroniques. L'intrus accède donc aux ports du serveur SMTP situés dans le réseau de l'entreprise en attaquant depuis le port source 25/TCP et peut atteindre par exemple le port telnet du serveur distant.

Il est important de noter que ce type d'attaque est rendu possible par l'absence de contrôle par l'équipement filtrant d'un ensemble de caractéristiques du paquet IP :

— aucune vérification des bits SYN et ACK n'est faite, ainsi le fait qu'un paquet SYN sans ACK arrive depuis Internet ne perturbe pas l'équipement filtrant qui est pourtant configuré pour n'accepter que les retours de sessions sortantes ;

— il n'y a pas de maintien de tables dynamiques de trafic ayant transité par l'équipement filtrant, et par conséquent celui-ci ne fait pas la différence entre une réponse à un trafic sortant et un trafic entrant initié de l'extérieur.

Ainsi, si l'équipement filtrant disposait d'une table dynamique indiquant les sessions sortantes en cours, et donc n'acceptait que les retours de sessions déjà en cours, ou simplement bloquait les paquets SYN sans ACK, alors ce type d'attaque ne serait plus possible.

■ **Routage à la source :** le routage à la source (*source routing*) fait partie intégrante du protocole TCP/IP. Il s'agit d'une option permettant à l'émetteur du paquet de spécifier le chemin réseau que doit emprunter le paquet pour arriver à destination.

Si le pare-feu n'a pas été configuré pour refuser les paquets ayant défini un routage à la source, alors il est possible de passer outre les règles de filtrage (et de routage). Nous analysons le détail de la technique du routage à la source au paragraphe 2.4.3.

■ **Modifier les attributs des paquets IP :** comme la technique citée précédemment, jouer sur les bits et les options des paquets IP envoyés au serveur peut permettre de traverser un équipement de protection selon ses capacités.

La méthode la plus simple consiste simplement à initier une session en activant le bit ACK dans le paquet IP. Certains équipements réseaux se contentent de bloquer les flux réseaux qui ont le bit SYN activé sans le bit ACK. Par conséquent, tout échange dont le bit ACK est actif est accepté. Bien sûr, le comportement réseau du système cible est également déterminant. Si ce système constate que tant SYN que ACK sont actifs, il doit refuser la connexion puisqu'il ne s'agit pas d'une session déjà initialisée. Mais la « bêtise » (ou la souplesse) de certains systèmes d'exploitation sur ce point ne cesse de surprendre [H 5832].

Il existe de multiples autres méthodes et leur dédier entièrement un document serait nécessaire. Certaines, comme la technique de l'arbre de Noël, vont jusqu'à activer tous les bits du paquet IP (SYN, ACK, PSH, URG...), et inversement, celle dite NULL envoie des paquets sans aucun bit actif.

■ **Fragmenter les paquets IP :** la technique la plus efficace pour traverser un pare-feu reste la fragmentation de paquets. Pourquoi la plus efficace ? Parce qu'elle fonctionne sur le plus grand nombre d'équipements filtrants. Il existe deux techniques de fragmentation : par petits paquets et par chevauchement.

● **Fragmentation par petits paquets :** le principe de la technique de fragmentation par petits paquets (*tiny fragments*) consiste à fragmenter l'en-tête du paquet IP afin que le filtre ne puisse pas déterminer s'il doit laisser passer le premier paquet ou pas sans possession du paquet suivant.

Le premier paquet contient alors des données telles que les ports source et destination ainsi que le numéro de séquence du paquet, alors que le second contient les bits du paquet (SYN actif et ACK inactif).

Dans les premières générations de pare-feu, le comportement appliqué au premier paquet était appliqué à tous les suivants. Ainsi, le pare-feu laissait passer le premier fragment (ne pouvant déterminer s'il fallait le bloquer ou pas) et donc le paquet suivant pouvait également passer. Cependant, sur le système cible visé, le paquet était réassemblé et permettait donc l'établissement d'une session non autorisée (figure **6**).

Fort heureusement, depuis que cette technique a été largement utilisée sur Internet, les constructeurs de solutions de pare-feu ont créé des contre-mesures pour éviter que cette attaque reste efficace.

● **Fragmentation par chevauchement :** la technique de fragmentation par chevauchement (*fragments overlapping*) repose sur le même principe que la précédente, mais avec deux différences :

— le paquet est découpé au moyen de l'option *overlapping*. Dans ce cas, le premier paquet contient les données de l'en-tête TCP avec les indicateurs inactivés, et le second paquet contient les données de la demande de connexion TCP ;

— les valeurs des *offsets* des fragments sont telles que les deux fragments se chevauchent au moment de leur assemblage sur le système cible et, par conséquent, les données du second fragment sont écrites par-dessus celles du premier (figure **7**).

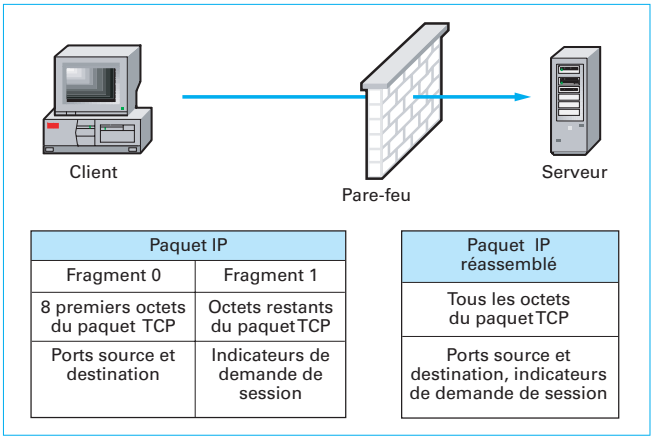


**Figure 6 – Fragmentation par petits paquets**

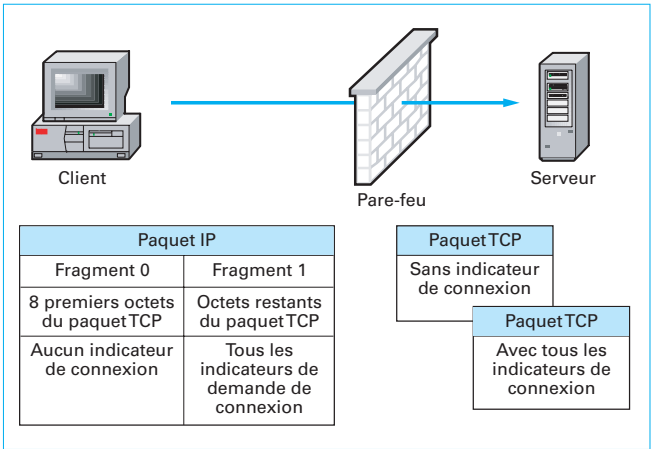


**Figure 7 – Chevauchement de fragments**

# 2. Récupérer des informations, modifier le comportement du réseau

L'intrus a maintenant pu établir une liste précise des différents systèmes d'exploitation présents sur un réseau en utilisant les techniques présentées au paragraphe 1. Cet intrus est maintenant désireux de déterminer quels sont les systèmes vers lesquels il doit porter son attention.

Malheureusement pour lui, nombreux sont les systèmes protégés par des mécanismes de sécurité et il doit se faire une idée de l'importance d'un système sans réussir une intrusion [H 5 840] sur chacun d'eux car cela prendrait trop de temps et surtout il serait détecté. Dans ce cas, la méthode la plus discrète et efficace reste de lire les données qui transitent sur le réseau. Cela s'appelle l'**écoute du réseau** (*sniffing*) .

Par ailleurs, nous constaterons qu'il est possible de modifier des informations en transit sur le réseau sans que cela soit détecté par les intervenants d'une transaction, tout comme il est possible de modifier le comportement du réseau, en forçant par exemple les chemins empruntés par les paquets de données.

Rappelons auparavant les techniques d'**échange d'information sur les réseaux Ethernet** :

— un réseau Ethernet peut être structuré de différentes manières : en bus ou en étoile ;

— les topologies en bus sont les vieilles technologies Ethernet [Ethernet fin (*thin*) et AUI (*thix*)] ;

— dans les topologies en étoile, l'étoile peut être directement reliée à d'autres ;

— de nos jours, la topologie en étoile est la plus fréquemment rencontrée.

Sur une technologie de type bus, les données sont déposées sur le bus et parcourent entièrement celui-ci pour atteindre la machine de destination. Par conséquent, quel que soit l'endroit sur le bus où se trouve une machine écoutant le réseau, celle-ci voit l'intégralité des données transitant sur le réseau.

Sur une technologie fonctionnant en structure étoilée, les données passent par un équipement réseau central, commutateur (*switch*) ou répéteur (*hub*). Si l'étoile est un répéteur, la donnée partant d'une machine A pour aller à une machine B est diffusée sur l'ensemble des liaisons Ethernet point à point (entre une machine et l'étoile).

Par conséquent, quel que soit le port sur lequel est présente une machine écoutant le réseau, celle-ci voit l'intégralité des données transitant sur le réseau.

Si l'étoile est un commutateur, alors les données entre la machine A et la machine B ne transitent que par les ports et câbles utilisés par ces deux machines.

## 2.1 Écoute du réseau

À la base, l'écoute du réseau (*sniffing*) n'est que l'utilisation d'une fonction standard d'Ethernet : le mode *promiscuous*. Au lieu de n'écouter que les données qui lui sont destinées, comme c'est le cas dans son fonctionnement normal, l'interface Ethernet d'un terminal dans le mode *promiscuous* s'intéresse à toutes les données en transit sur le lien. Si la topologie du réseau est de type bus, alors il suffit d'activer l'interface réseau de la machine en mode *promiscuous* pour voir toutes les données en transit sur le réseau. Des outils spécialisés d'analyse de protocoles tels que Sniffer Pro, tcpdump, windump ou des sondes RMon par exemple, permettent alors de simplifier l'analyse pour n'extraire que l'information désirée. En revanche, lorsque le réseau s'appuie sur une étoile commutée, les choses deviennent un peu plus difficiles.

## 2.2 Attaque d'un commutateur

Un commutateur a pour fonction de ne fournir la donnée qu'au port hébergeant la machine destinatrice. Pour cela, le commutateur gère une table dynamique des adresses MAC physiques des machines situées sur chaque port. Lorsque la donnée part d'une machine A vers une machine B, le commutateur reçoit en fait un paquet destiné à une adresse MAC particulière. Il regarde dans sa table et envoie le paquet au bon port physique sur la bonne interface. Si le commutateur ne trouve pas l'adresse MAC dans sa table, il interroge alors tous les ports physiques (*via* une requête ARP) pour tenter de trouver lequel est le bon. Un intrus branché sur un commutateur ne doit voir que :

— les données qui sont destinées à une machine branchée sur le même port que lui, donc lui-même ou une autre s'il est branché sur un répéteur branché au commutateur ;

— les données envoyées sous forme de *broadcast* ou *multicast*. Dans ce cas, le commutateur envoie les données à tous les ports hébergeant des machines.

De plus, le commutateur sait offrir d'autres services tels que le compartimentage de réseaux locaux au sein du même commutateur. Cette technologie appelée **VLAN** (Virtual LAN) permet donc à une entreprise de créer des réseaux virtuels distincts sur une

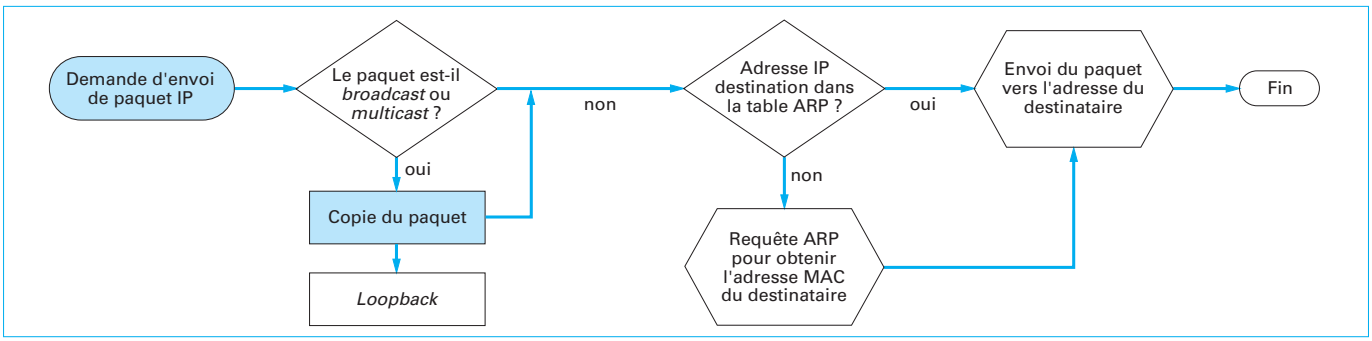


**Figure 8 – Émission d'un paquet IP sur une interface Ethernet**

même infrastructure ; ils peuvent être spécialisés et comprendre par exemple un réseau comptabilité, un réseau production et un réseau bureautique. Ainsi, parce que le commutateur se trouve être le point de relation avec des réseaux qui pourraient ne pas être liés logiquement au réseau d'entreprise, par exemple, il devient une cible privilégiée s'il est possible d'écouter ou d'accéder à ces autres réseaux par l'exploitation d'une faiblesse permettant de passer outre le compartimentage VLAN.

Bien sûr, le commutateur, qui devient alors la cible de l'intrus, peut être mal configuré et l'intrus peut user de ce type de faiblesse pour en prendre le contrôle. Si cela arrive, l'intrus peut alors faire ce que bon lui semble sur cet équipement réseau. Cependant, nous n'aborderons pas cette méthode car il s'agit davantage d'une intrusion contre un système [H 5 832]. Nous resterons ici au niveau réseau et par conséquent aux techniques qui ne demandent pas la prise de contrôle d'un équipement.

L'intrus doit donc attaquer le commutateur au niveau réseau (ou plutôt liaison de données) afin d'avoir la visibilité sur les données qui ne lui sont pas destinées et de casser le concept de réseau virtuel logique. Il dispose pour cela de plusieurs techniques, dont seulement quelques-unes sont présentées ci-après. On peut encore citer maintes autres méthodes telles que l'attaque par génération de paquets aléatoires (*random frame stress attack*), l'attaque par famine DHCP (*DHCP starvation attack*), l'attaque par force brute *multicast* (*multicast brute force attack*), l'analyse de redondance (*failover analysis*), les attaques contre VTP (VLAN Trunk Protocol), VPMS/VPQ (VLAN Policy Management Server/VLAN Query Protocol), CDP (Cisco Discovery Protocol), PVLAN (Private VLAN), etc.

### 2.2.1 Techniques d'empoisonnement ARP

Les techniques dites d'empoisonnement ARP (*ARP poisoning*) ont pour but de corrompre la table ARP (Address Resolution Protocol) d'un commutateur à travers deux méthodes :

— l'envoi de fausses réponses à des requêtes ARP (*ARP request*) ;
— l'envoi de paquets avec une adresse MAC usurpée.

Ces techniques d'empoisonnement sont présentées ici contre un commutateur, mais elles fonctionnent avec plus ou moins de succès contre tout équipement branché sur un réseau TCP/IP.

■ **Envoi de fausses réponses aux requêtes ARP :** lorsqu'un terminal branché sur un réseau désire envoyer un paquet de données à un autre, il a besoin de l'adresse MAC du récepteur dudit paquet. Si ce récepteur est situé dans le même sous réseau, alors la machine émettrice se contente d'émettre une requête ARP afin d'obtenir l'adresse MAC du récepteur. Cette adresse est ensuite stockée dans une table ARP et associée à un port particulier du commutateur. Ainsi, lorsque le commutateur doit envoyer un paquet de données, il optimise son comportement en s'appuyant sur cette table pour déterminer le port où déposer le paquet. Sur tout équipement relié à un réseau TCP/IP, l'envoi d'un paquet se passe comme le montre la figure **8**.

Le but de l'intrus est donc soit de recevoir des données destinées à une adresse IP (et donc en fait une adresse MAC) située dans le même VLAN que lui mais qui ne devraient pas être envoyées sur le port où il est connecté, soit de passer outre le concept de compartimentage que constitue le réseau virtuel logique afin de recevoir des données provenant en plus d'un autre réseau virtuel présent sur le même commutateur.

Pour cela, la machine de l'intrus envoie des réponses ARP (*ARP reply*) au commutateur afin de lui faire croire que l'adresse MAC dont il désire recevoir les données est bien présente sur le port. Bien sûr, cela suppose que l'intrus ait pu connaître ladite adresse MAC d'une manière ou d'une autre.

Selon le constructeur et la configuration du commutateur, ce dernier se comporte de très différentes manières (figure **9**). Le commutateur peut en effet accepter d'envoyer les paquets destinés à la machine réellement sur le port ainsi qu'à celle que l'intrus veut écouter, ou ignorer la réponse qui lui semble anormale sur ce port. Mais si le commutateur présente des faiblesses de programmation dans son algorithme de traitement, alors il peut se trouver en situation de déni de service, ignorer l'attaque, envoyer les paquets destinés à l'adresse MAC sur le port de l'intrus uniquement ou envoyer ces données sur chaque port, selon que le commutateur remplace ou ajoute l'adresse MAC dans sa table ARP et l'associe au port de l'intrus qui contient l'adresse MAC usurpée.

Bien sûr, détecter ce type d'attaque est très simple lorsque l'on surveille les traces (*logs*) du commutateur. Celui-ci note en général la duplication de l'adresse MAC sur ses ports ou la présence de plusieurs adresses MAC sur un même port. Enfin, des outils tels que *ARPwatch* ont pour fonction de détecter les couples d'adresses IP/MAC et il est facile de remarquer qu'une même adresse IP utilise plusieurs adresses MAC *via* cet outil si la machine chargée de cette tâche reçoit toutes les données passant par le commutateur ou par l'examen des tables ARP des équipements. La plupart des commutateurs offrent ce service.

■ **Envoi de paquets usurpant une adresse MAC :** avec le temps, les commutateurs sont devenus de plus en plus « intelligents ». Ils sont maintenant capables de s'autoparamétrer et donc d'ajuster leur configuration en fonction du comportement de leurs ports. Ainsi, alors qu'il fallait auparavant spécifier si un port hébergeait un répéteur ou non, le commutateur s'adapte maintenant tout seul lorsqu'il détecte plusieurs adresses MAC sur un même port.

Une autre méthode dite **attaque de raisonnement** (*reasoning attack*) consiste donc à envoyer des paquets en usurpant directement une adresse MAC. Le commutateur détecte alors ces paquets et « pense » que cette adresse est présente sur son port. Les comportements possibles sont identiques à ceux de la méthode précédente avec la variante de la présence de l'algorithme (figure **10**).

Toutefois, ce type d'attaque peut offrir des possibilités supplémentaires par rapport à la méthode précédente. Ainsi, une extension de cette attaque consiste à renvoyer les données obtenues vers le commutateur en continuant à usurper l'adresse MAC. Cer-

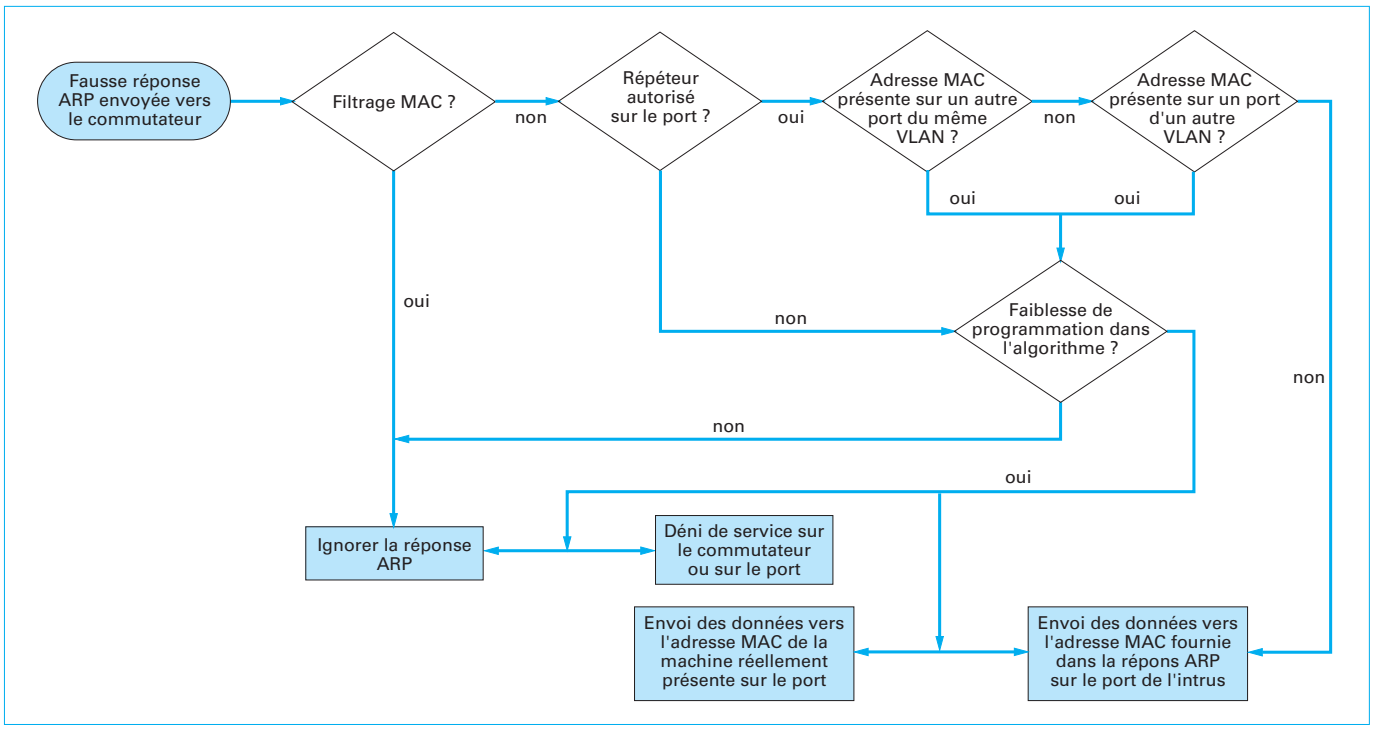


**Figure 9 – Comportements possibles du commutateur en présence de fausses adresses ARP**

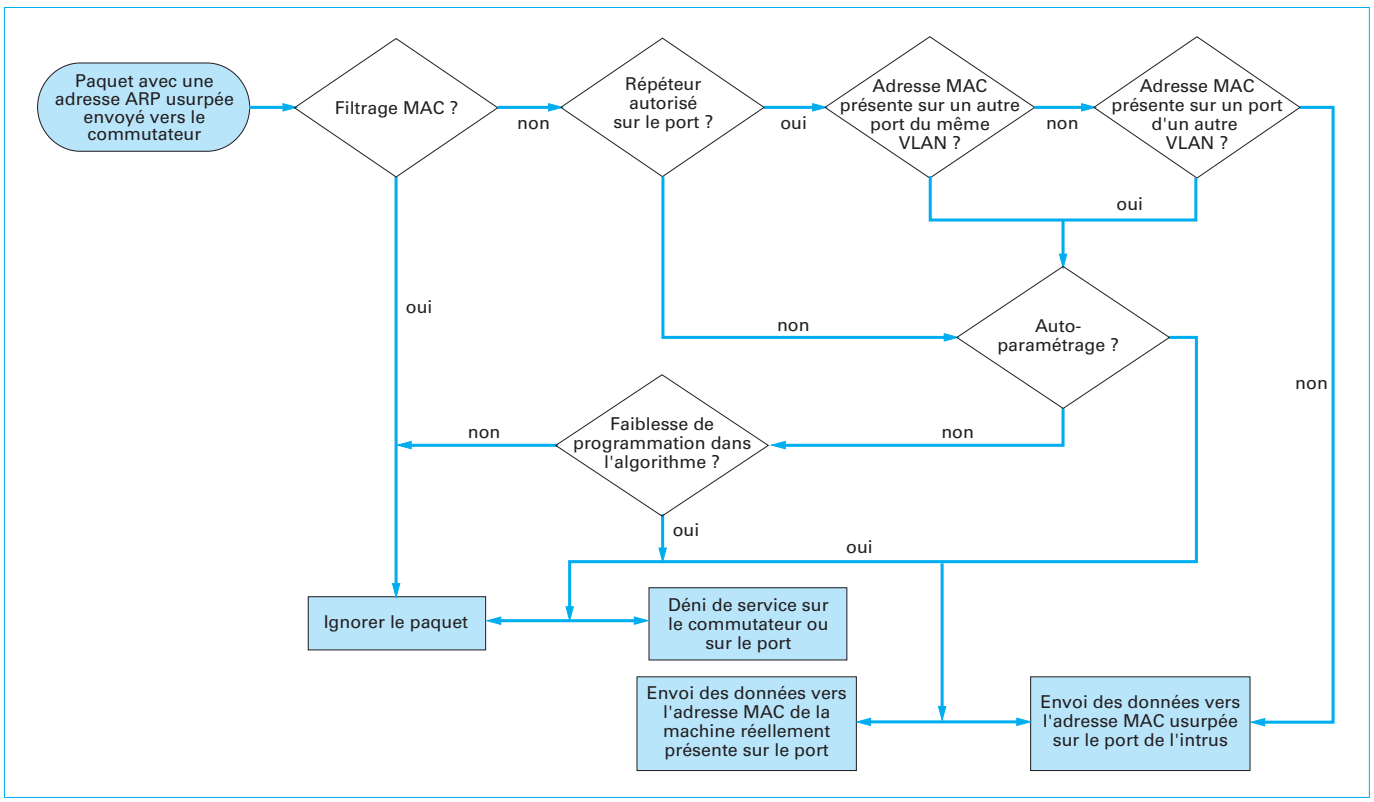


**Figure 10 – Comportements possibles du commutateur en présence d'adresses MAC usurpées**

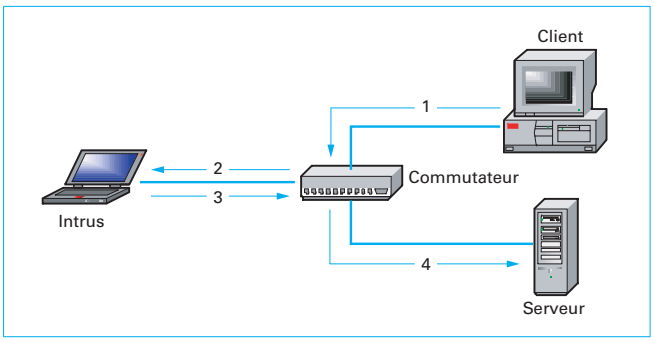


**Figure 11 – Comportement du commutateur attaqué**

tains commutateurs refusent en effet de renvoyer le paquet vers le port d'où il vient. S'il se comporte ainsi, le commutateur renvoie alors le paquet usurpé vers les autres ports qui contiennent également cette adresse MAC et l'intrus peut alors se positionner pour une « attaque de l'homme au milieu » (§ 2.4) et donc se trouver un point de passage obligatoire entre le client et le serveur.

Au départ, le client et le serveur s'échangent normalement des données par le chemin « client, commutateur, serveur » et inversement, puis l'intrus envoie des paquets usurpant les adresses MAC du serveur et du client. À ce moment, le comportement du commutateur change (figure **11**) :

1. le client demande la connexion avec le serveur. Le commutateur reçoit alors un paquet du client destiné au serveur ;
2. le comportement du commutateur n'est plus normal. Celui-ci envoie donc le paquet vers la machine de l'intrus ;
3. l'intrus copie le paquet et le renvoie alors vers le serveur avec l'adresse MAC du client ;
4. le commutateur ne peut se résoudre à renvoyer le paquet vers le port d'où il vient et le renvoie donc vers le port du serveur.

La réponse se fait alors selon le procédé inverse. La machine de l'intrus se trouve parfaitement en position de point de passage obligatoire et donc bien dans une situation d'attaque de l'homme au milieu. Il faut cependant qu'elle continue constamment à envoyer des paquets usurpant les adresses MAC du serveur et du client pour rester dans cette position.

Il est important de noter que ce comportement n'a pu être reproduit avec succès que sur un seul type de commutateur sur un grand nombre d'équipements testés.

## 2.2.2 Attaques basées sur le protocole 802.1q

Le protocole 802.1q a pour fonction de permettre à des commutateurs de s'échanger des données, principalement afin de servir des VLAN partagés sur plusieurs commutateurs. Normalement, seuls les ports destinés à permettre aux commutateurs de communiquer (appelés ports *trunk*) sont explicitement paramétrés pour accepter de travailler selon le protocole 802.1q. Cependant, en fonction de la configuration du commutateur, il peut être possible d'utiliser ce protocole sur un port non *trunk*, surtout si le commutateur est configuré pour faire du DTP (Dynamic Trunk Protocol). Le rôle du protocole DTP est de permettre à n'importe quel port de recevoir un commutateur sans qu'il soit nécessaire de configurer le port en mode *trunk*. Tout se fait automatiquement.

■ **Saut de VLAN :** cette technique, appelée aussi *VLAN hopping*, consiste à envoyer des paquets 802.1q ou ISL (Inter Switch Link) sur un port du commutateur afin de l'utiliser comme un port *trunk*.

Ainsi que l'illustre la figure **12**, il devient alors possible pour l'intrus, lorsque l'attaque réussit, d'écouter l'intégralité du trafic passant sur le commutateur et non pas seulement celui du VLAN associé au port attaqué, puisqu'un port *trunk* est membre de tous les VLAN.

■ **Saut de VLAN par double encapsulation 802.1q :** cette technique, appelée aussi *double encapsulation 802.1q VLAN hopping* ou *double tagged 802.1q VLAN hopping*, consiste à envoyer des trames vers une machine située dans un autre VLAN, sans risquer de mettre le commutateur en déni de service, effet secondaire regrettable possible avec les attaques basées sur l'usurpation d'adresses MAC.

Dans ce type d'attaque, l'intrus envoie un paquet avec deux en-tête 802.1q. Comme le montre la figure **13**, le premier en-tête

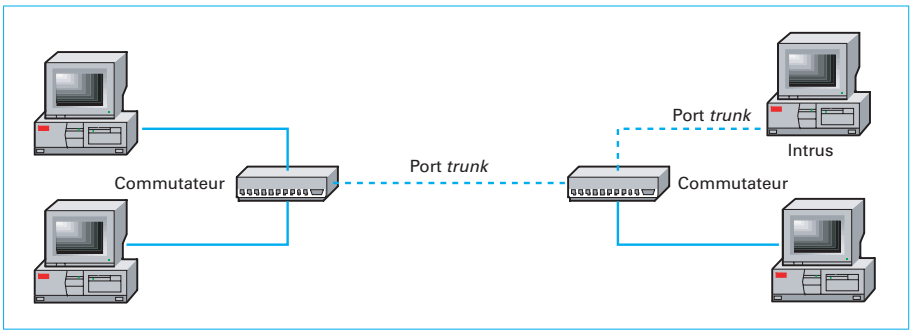


**Figure 12 – Saut de VLAN**

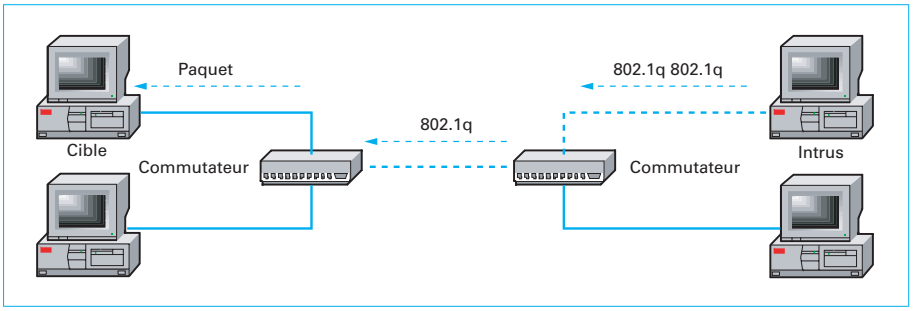


**Figure 13 – Saut de VLAN par double encapsulation 802.1q**

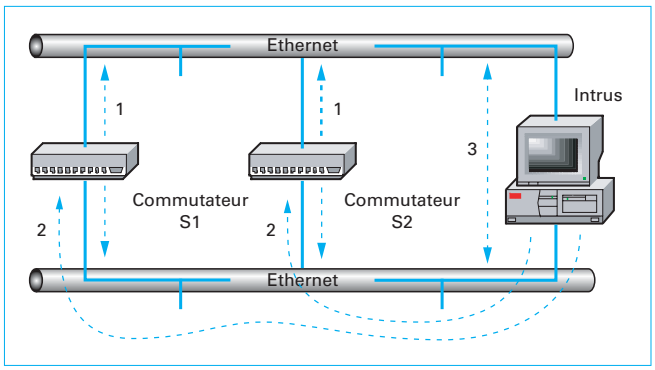


**Figure 14 – Attaque STP**

est ôté au premier commutateur rencontré. Le paquet est ensuite véhiculé jusqu'au dernier commutateur qui retire alors l'en-tête restant pour envoyer les données vers la machine cible. Il faut noter que cette technique d'attaque ne fonctionne que dans une direction.

### 2.2.3 Attaques sur le protocole Spanning Tree

Le protocole Spanning Tree (STP) a pour fonction la gestion de la topologie du réseau et assure ce service en fournissant des chemins redondants et en évitant les boucles de routage. Ce protocole est intégré dans le niveau 2 du modèle OSI (couche liaison de données) et fait partie des services assurés par les commutateurs. Il utilise un algorithme distribué qui sélectionne un pont d'un réseau connecté de manière redondante comme la racine d'un arbre associé à la topologie courante. STP s'appuie sur l'envoi de messages BPDU (Bridge Protocol Data Unit) entre les commutateurs pour les échanges d'ordres et de données. Les membres d'une infrastructure STP sont des commutateurs tous reliés directement entre eux.

Afin d'éviter les conflits, chaque commutateur assure un rôle dans l'infrastructure STP. Parmi ces rôles, un seul nous intéresse : celui appelé *root* (racine). En effet, ce commutateur a la charge de donner les directives STP à tous les autres commutateurs de l'infrastructure et peut donc changer le comportement du réseau. Les messages BPDU sont utilisés entre les commutateurs afin d'élire dans le réseau celui qui sera *root*.

Le principe de l'attaque sur STP consiste donc à envoyer des BPDU avec des valeurs qui font de l'émetteur le commutateur *root* du réseau afin que tous les paquets lui soient envoyés.

Comme le montre la figure **14**, les commutateurs S1 et S2 sont normalement en charge du transit des données entre les deux réseaux (étape 1). Cependant, l'intrus a réussi à se faire élire commutateur *root* grâce à une attaque STP (étape 2). Il devient donc le goulot d'étranglement entre les deux réseaux (étape 3).

## 2.3 Usurpation d'adresse IP

Le pirate a découvert une machine particulièrement intéressante (serveur) qui est protégée pour ne permettre qu'à certaines adresses IP (parmi celles-ci figure celle du client) d'accéder à ses ressources. L'intrus doit donc avoir recours à de nouvelles techniques d'attaque puisque ARP ne peut aider dans ce cas.

Parmi ces techniques, l'usurpation d'adresse IP (*IP spoofing*) reste très utilisée sur Internet, preuve de son efficacité. Le principe en est très simple : il consiste à émettre des paquets forgés avec une adresse IP qui n'appartient pas à la machine émettrice du paquet. Ce type d'attaque sert principalement dans trois cas :

— masquer l'adresse IP d'une machine au sein d'un grand nombre d'autres adresses. Il s'agit alors de créer des leurres afin

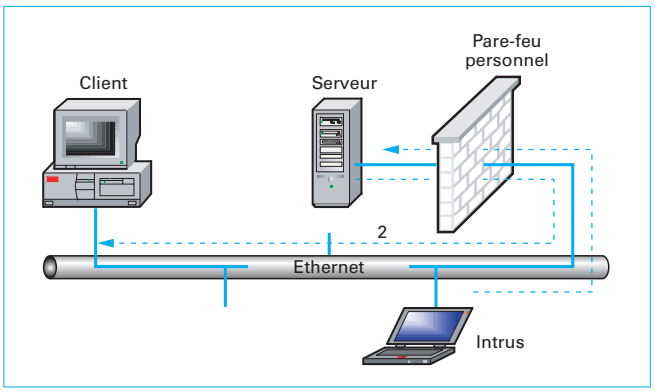


**Figure 15 – Réponse face à une usurpation d'adresse IP**

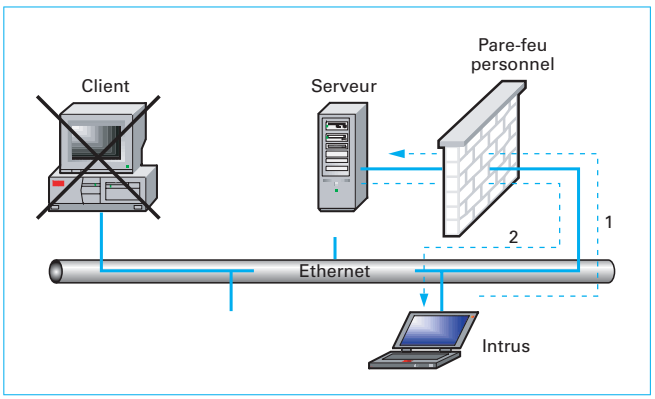


**Figure 16 – Usurpation d'adresse IP complète**

que la victime perde du temps à trouver la machine attaquante, voire ne puisse le faire ;

— permettre à la machine attaquante d'envoyer des paquets de données qui sont autorisés à passer un mécanisme de sécurité filtrant les adresses sources ;

— si la machine attaquante est placée logiquement entre deux machines (attaques dites de l'homme au milieu), lui permettre de se placer en relais transparent, avec toutes les possibilités présentées au paragraphe 2.4.

Nous nous concentrons ici sur l'usurpation d'adresse dans le but de passer outre un mécanisme filtrant les adresses IP sources.

L'intrus génère donc des paquets en usurpant l'adresse IP du client afin de franchir le mécanisme de sécurité (un pare-feu personnel protégeant le serveur). Pour cela, il doit auparavant déterminer quels sont les numéros de séquence des paquets du serveur par l'envoi de paquets et l'analyse de l'algorithme d'incrémentation du serveur. Heureusement, les constructeurs des systèmes d'exploitation ont mis en place des algorithmes d'incrémentation de plus en plus complexes afin de lutter contre ce type d'attaque. Ils n'ont cependant pas encore pris l'habitude d'activer ces algorithmes par défaut. Nous considérons ici que le numéro de séquence est aisément prévisible afin que l'intrus puisse mener à bien son attaque (figure **15**).

Sur la figure **16**, nous voyons le trafic du réseau tel qu'il devrait se produire :

1. le paquet forgé avec une adresse usurpée est envoyé vers le serveur et l'atteint puisque le pare-feu personnel accepte le trafic provenant de cette adresse IP source ;
2. le serveur répond alors à la demande en renvoyant un paquet vers le client qu'il pense être l'émetteur du paquet.

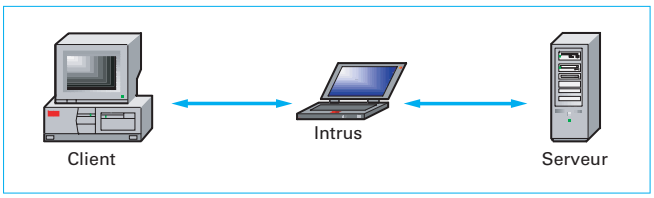


**Figure 17 – Relais applicatif**

Nous constatons donc que le client reçoit la réponse à la demande de session faite par l'intrus en son nom. En recevant ce paquet, le client peut lui-même répondre au serveur et faire échouer cette tentative d'usurpation. Pour éviter cette situation, il est nécessaire de rendre inopérante la machine dont on usurpe l'adresse IP. Comme nous le verrons au paragraphe 3 sur les dénis de service et dans [H 5832] qui traite des attaques des systèmes, il existe de multiples méthodes pour « tuer » le client.

Sur la figure **16**, on constate que le client est rendu incapable de converser avec le réseau. La machine de l'intrus peut alors répondre et établir une véritable session avec le serveur en tant que client, trompant ainsi complètement le mécanisme de sécurité.

## 2.4 Attaques dites de l'homme au milieu

Nous avons vu (§ 2.3) qu'il était possible pour un intrus de se faire passer pour un équipement présent sur le réseau. Malheureusement, dans la plupart des cas, il est probable que la machine ainsi abusée perde la possibilité d'échanger des informations avec le réseau, ce qui signifie que ces attaques pourraient être rapidement détectées.

Un pirate, pour être efficace, doit rester discret et indétecté, voire indétectable. Pour cela, la méthode idéale est de se placer entre deux machines comme un équipement normal du réseau (tel qu'un routeur par exemple).

Lorsqu'une attaque place le pirate entre le client et le serveur, elle est dite de l'homme au milieu (*man in the middle*). Il existe différentes architectures pour cette attaque.

### 2.4.1 Relais applicatif

Le relais applicatif n'est pas à proprement parler une technique d'attaque, mais plutôt une « architecture » que l'intrus désire obtenir pour augmenter l'efficacité et la discrétion de son piratage. Cette architecture présente l'énorme avantage de ne pas obliger le pirate à se trouver logiquement entre le client et le serveur. Le principe du relais applicatif est obligatoire lorsque le pirate a besoin de décoder des flux chiffrés entre deux machines.

L'intrus cherche donc à se placer dans les flux circulant entre le client et le serveur afin d'avoir la visibilité de tous les flux en transit, mais également afin d'être à même de modifier ces flux (figure **17**).

Prenons par exemple le cas de l'espionnage de flux HTTPS entre client et serveur. L'intrus, ne pouvant espérer déchiffrer une clé de 128 bits (par exemple) dans un délai raisonnable, peut préférer une meilleure tactique qui consiste à se placer sur le flux entre le client et le serveur. L'intrus joue alors le rôle de relais applicatif (*proxy*) HTTPS.

Sur la figure **18**, nous voyons le trafic du réseau tel qu'il se produit :

1. le client envoie sa requête HTTPS vers l'intrus qui n'est pas du tout sur le chemin logique entre le client et le serveur HTTPS. Il négocie donc sa session chiffrée avec lui ;
2. l'intrus déchiffre la requête et la renvoie vers le serveur HTTPS. Il négocie donc sa session avec le serveur ;
3. le serveur HTTPS répond à l'intrus ;
4. l'intrus répond au client.

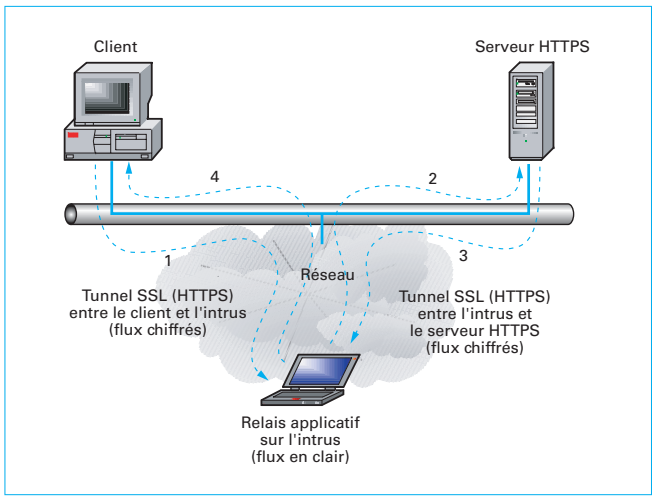


**Figure 18 – Relais applicatif HTTPS**

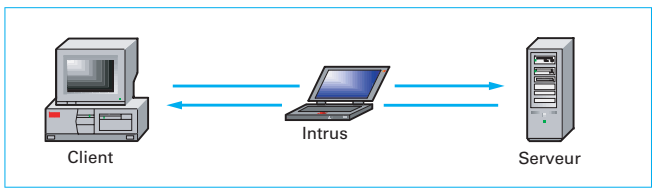


**Figure 19 – Relais transparent**

On constate alors que :

— l'intrus dispose des flux HTTPS en clair (non chiffrés) ;

— l'intrus peut modifier la demande HTTPS renvoyée vers le serveur HTTPS ;

— l'intrus peut falsifier les réponses renvoyées par le serveur HTTPS vers le client.

Cette architecture de relais applicatif peut aisément être reproduite pour tous les services du réseau qui sont naturellement bâtis pour être relayés tels que HTTP, HTTPS, SMTP... Notons tout de même que la discrétion n'est pas encore idéale car tant le client que le serveur HTTPS peuvent facilement se rendre compte qu'ils ne se parlent pas directement. Il suffit donc au serveur HTTPS de comparer l'adresse de provenance des paquets du client et l'adresse source indiquée dans les requêtes (par un simple programme Java par exemple) pour découvrir la supercherie, et de même pour le client.

### 2.4.2 Relais transparent

Ici, le pirate peut être complètement indétectable car il intercepte le trafic du client et le réachemine vers le serveur en tant que client par usurpation de son adresse IP. Bien sûr, cela implique que le client soit logiquement sur le chemin réseau entre le client et le serveur. Ce type d'architecture doit donc être associé à des attaques visant à modifier le comportement du réseau telles que les attaques contre le commutateur (§ 2.2) ou contre le routage (§ 2.4.3).

Par ailleurs, la technicité à mettre en œuvre ici est bien plus complexe que dans le cas du relais applicatif. En effet, il ne s'agit plus de relayer des requêtes qui sont prévues pour être relayées, mais de se comporter comme un équipement du réseau transparent (dans le cas le plus simple) et de modifier les données à la volée dans le cas le plus difficile et le plus fréquent.

Sur la figure **19**, nous voyons que l'intrus a réussi à se placer dans la situation de relais transparent entre le client et le serveur.

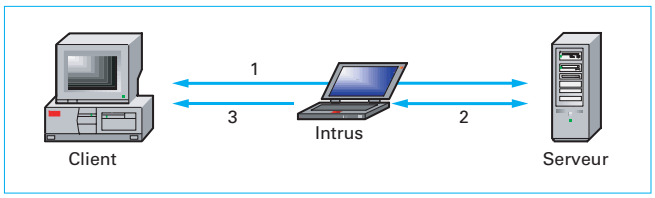


**Figure 20 – Détournement de la session entre le client et le serveur**

Il reçoit donc toutes les données qui transitent entre ces deux machines. Il peut écouter ce trafic mais il peut également le modifier. En effet, l'intrus dispose de toutes les informations nécessaires comme le port source, le numéro de séquence, etc. Il est donc à même de modifier le contenu d'un paquet avant de le réacheminer vers le serveur en respectant tous les en-têtes des paquets.

Cependant, l'intrus peut vouloir faire mieux que de modifier à la volée les données échangées. En effet, les modifications engendrées peuvent être détectées immédiatement par le client, qui peut constater une différence entre la réponse fournie et celle attendue, ou un affichage erroné des données (dans le cas d'une session telnet par exemple). Le détournement de session permet à l'intrus d'échanger des données avec le serveur sans que ni le serveur ni le client ne s'en aperçoivent.

L'intrus peut utiliser une nouvelle arme : le **vol de session** (*hijacking*). Dans cette technique d'attaque reposant sur une architecture en relais transparent, l'intrus interrompt la session entre le client et le serveur, puis se fait passer pour le client et continue la session qui avait été ouverte par le client.

Sur la figure **20**, nous voyons le client en session sur le serveur (étape 1). L'intrus est en position de relais transparent et se contente de réacheminer les paquets. Dans l'étape 2, l'intrus décide de voler la session. Il arrête donc de répondre au client ou lui renvoie des réponses qui le satisfont et continue la session établie avec le serveur en tant que client. Enfin, dans l'étape 3, l'intrus termine proprement (afin de réduire les soupçons de piratage) la session que le client avait avec le serveur. Pour cela, l'intrus renvoie au client une demande de fin de session (paquet FIN) en tant que serveur. Le client voit sa session interrompue et pense qu'il s'agit d'un problème ponctuel du réseau. Soit il arrête alors de travailler, soit il relance une nouvelle session que l'intrus se contente de réacheminer de manière transparente, puisqu'il dispose maintenant d'une session rien que pour lui.

Voir une session interrompue de cette manière brutale est courant. s'interroge-t-on sur le motif réel de cette interruption ?

## 2.4.3 Modification du routage

Afin de pouvoir se placer en situation de relais transparent, l'intrus doit modifier le comportement du réseau pour être considéré comme un point de passage entre deux machines.

Nous avons vu (§ 2.2) comment modifier le comportement du réseau lorsque l'intrus est placé sur le réseau local grâce aux attaques contre le commutateur ou l'usurpation ARP.

Cependant, lorsque l'intrus se trouve en dehors du réseau local, ce type d'attaque est plus difficile à mener. Il faut alors s'attaquer au routage des paquets qui permet de franchir le concept de réseau local au profit du réseau global.

Nous n'abordons ici que les attaques de routage visant des protocoles simples de routage. Les attaques vers les protocoles avancés (OSPF, BGP...) sont traités au paragraphe 2.6.

■ **Routage à la source :** la technique du routage à la source (*source routing*) est des plus simples à mettre en œuvre puisqu'elle est définie dans les normes du protocole TCP/IP. Le principe de cette attaque est que le pirate force le routage que doit suivre le paquet pour atteindre le serveur et revenir vers le client. Pour cela, il inclut dans ses paquets ce chemin grâce au routage à la source. Ainsi, le pirate peut usurper une adresse IP (même si celle-ci est dans le réseau local où est situé le serveur).

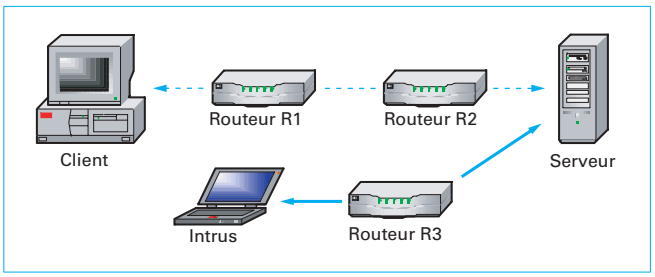


**Figure 21 – Routage à la source**

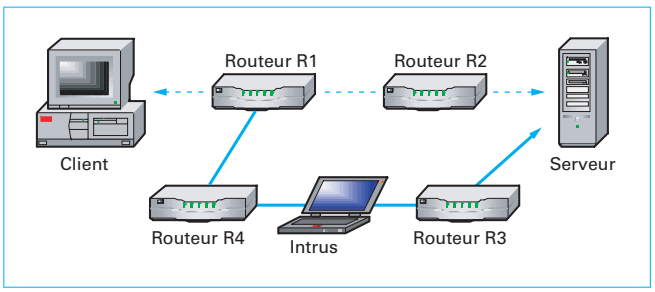


**Figure 22 – Modification du routage par ICMP Redirect**

Dans l'exemple de la figure **21**, nous voyons que le client échange des données avec le serveur. Normalement, cet échange devrait se faire en passant par les routeurs R1 et R2. Mais l'intrus envoie ses paquets vers le serveur en usurpant l'adresse IP du client ainsi qu'en spécifiant un routage à la source (passage obligatoire par le routeur R3). Le serveur reçoit alors les paquets de l'intrus et renvoie les réponses en respectant le routage fourni (routeur R3), lui permettant ainsi d'établir une session en se faisant passer pour le client.

■ **Attaque avec ICMP Redirect :** une autre méthode pour modifier le routage s'appuie sur le type Redirect du protocole ICMP. Le principe consiste à convaincre le réseau (les routeurs ou les systèmes présents sur le réseau) que le meilleur chemin n'est pas celui qu'il croit, mais plutôt celui qui passe par l'intrus.

Ainsi, l'intrus (figure **22**), placé en position de relais transparent, peut écouter ou modifier tout le trafic qui passe entre les routeurs R3 et R4. Afin de pouvoir faire de même pour le trafic transitant entre le client et le serveur, l'intrus envoie au serveur des paquets ICMP Redirect pour le convaincre que le routeur R3 est le meilleur chemin pour parler avec le client. L'intrus peut aussi procéder autrement en envoyant au routeur R1 des paquets ICMP Redirect afin de le convaincre que passer par le routeur R3 est le meilleur chemin pour atteindre le serveur.

■ **Attaques sur IRDP :** le protocole IRDP (ICMP Router Discovery Protocol) est utilisé pour indiquer à toute machine quelle est la route par défaut. Une technique d'attaque permet de changer la valeur de la route par défaut d'un réseau. Ainsi que le présente la figure **23**, l'attaque se déroule en quatre étapes.

En premier lieu, nous constatons que le client utilise le routeur comme sortie par défaut. Ensuite, l'intrus envoie des paquets de mise à jour IRDP sur le réseau. Puis, il réussit à rendre le routeur inopérationnel suite à un déni de service, ceci pour éviter qu'il n'émette à son tour des mises à jour IRDP. Il dispose pour cela de multiples solutions ainsi que nous allons le voir dans le

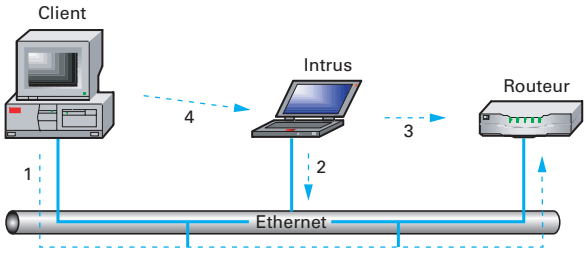


**Figure 23 – Attaque IRDP**

paragraphe 3 qui traite des techniques pour engendrer des dénis de service. Enfin, la quatrième et dernière étape où le routeur assurant le routage par défaut est mis hors service fait que la machine de l'intrus devient la route par défaut, le plaçant ainsi en position de relais transparent.

## 2.5 Attaques *wireless*

La technologie sans fil (*wireless*), aussi appelée IEEE 802.11 (suivi de différentes lettres selon la version) [TE 7 377], est de plus en plus présente dans notre vie quotidienne. Par le confort qu'elle procure en permettant un échange de données à haute vitesse comparable à celui de la technologie Ethernet mais en étant affranchi de la contrainte de la liaison physique filaire, elle a suscité un engouement parmi les utilisateurs qui se sont empressés de s'équiper. Mais personne ne s'est préoccupé de la sécurité de cette technologie comme c'est trop souvent le cas, même si tout le monde a bien compris que la « bulle » d'accès qu'une telle installation offrait pouvait être accessible à des personnes non autorisées. Avant toute chose, il faut bien appréhender ce concept de « bulle ».

Lorsqu'un point d'accès sans fil est installé, il peut être accédé par toute personne se trouvant dans la zone d'émission/réception du signal radio. Cette zone dépend de la puissance du point d'accès mais également de l'environnement physique de celui-ci. En effet, la zone se trouve réduite si les ondes doivent traverser des obstacles tels que des murs. Ainsi, la zone de portée d'un point d'accès de campus universitaire se trouve être réellement comme l'illustre la figure **24**. La zone de portée s'étend à travers plusieurs blocs de maisons grâce aux murs qui permettent aux ondes de rebondir.

Pour détecter si l'on se trouve dans la zone de couverture d'un point d'accès, il suffit donc d'écouter le réseau (comme cela se fait avec d'autres technologies telles qu'Ethernet). Il faut donc s'équiper d'un ordinateur portable, d'une carte sans fil de la génération la plus récente (ces cartes étant en général compatibles avec les générations précédentes de vitesse inférieure) et du logiciel approprié. C'est la panoplie nécessaire du *war driving*.

Le *war driving* consiste à conduire ou marcher en ville en écoutant le réseau sans fil afin de repérer les points d'accès. Il a de plus été inventé une signalétique associée à une pratique, appelée le *war chalking*. Cette pratique consiste à relever l'existence d'un point d'accès aux environs sous la forme d'un signalement fait à la craie (*chalk*). On peut ainsi rencontrer sur les murs des villes les signes présentés dans le tableau **1**.

Par ailleurs, les pirates emploient des astuces pour augmenter la zone d'émission/réception en émettant de manière plus unidirectionnelle (la technologie sans fil étant souvent émettrice sur 360°). Ces « astuces de collégien » sont très efficaces, même si elles ne relèvent pas vraiment de la haute technologie. La figure **25** présente les méthodes les plus utilisées, même en pleine action, en raison de leur faible coût. Ces méthodes restent aisées à utiliser, même en pleine action.

**Nota :** je déconseille cependant de passer devant la maréchaussée en conduisant ainsi.


**Figure 24 – Zone de portée réelle d'un point d'accès sans fil**

**Tableau 1 – Signes du *war chalking***

| | |
|---|---|
| SSID<br>bande passante | **Réseau sans fil ouvert**<br>Il y a ici un réseau sans fil ouvert au public, même si celui-ci n'est pas autorisé à l'utiliser. La sécurité de ce réseau est si faible que quiconque peut y accéder.<br>Le SSID et la bande passante sont indiqués. |
| SSID | **Réseau sans fil fermé**<br>Il y a ici un réseau sans fil qui est fermé d'accès au public. Tout ce qui a pu être identifié de ce réseau est son SSID, et accéder à ce réseau n'est pas facile.<br>Le SSID est indiqué. |
| SSID contact<br>W<br>bande passante | **Réseau sans fil protégé par WEP**<br>Il y a ici un réseau sans fil s'appuyant sur le chiffrement WEP pour se protéger (1).<br>Le SSID, le nom du propriétaire de l'accès et la bande passante sont indiqués. |

(1) Malheureusement pour le propriétaire, le chiffrement WEP est très facile à pirater comme nous allons le voir plus loin.

À titre d'exemple, la figure **26**, dont le titre pourrait devenir celui d'un film policier, montre le résultat d'un *war driving* effectué sur l'artère principale de la Silicon Valley située au sud de San Francisco en Californie, berceau de l'informatique mondiale. Un grand nombre de points d'accès ont été détectés, et la plupart étaient insécurisés.

Une fois qu'un intrus est dans la zone de portée d'un point d'accès et si le point d'accès est totalement insécurisé, il lui suffit d'établir une connexion pour y accéder. Nous n'abordons pas ce sujet car cela ne nécessite aucune méthode ou technique particulière ; il suffit d'établir la connexion pour accéder au réseau ainsi configuré. Si le point d'accès est sécurisé, l'intrus dispose d'une gamme de techniques d'attaque.

■ **Usurpation d'adresse MAC :** nous avons abordé les attaques s'appuyant sur une adresse MAC usurpée (§ 2.2.1). Ce type d'attaque est également applicable à la connectivité sans fil. Il est même plus simple car si la présence de deux adresses MAC identiques sur un commutateur peut engendrer des perturbations dans l'échange de paquets ou au niveau du commutateur, cette problématique n'apparaît pas au niveau d'un point d'accès sans fil car il envoie les données sur des ondes radio et ne peut connaître le nombre d'émetteurs/récepteurs utilisant cette adresse MAC, du moment qu'ils sont

**Figure 25 – Deux méthodes pour augmenter la distance d'émission/réception du signal**

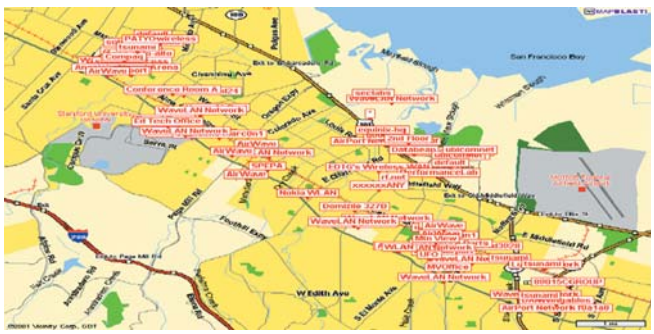

**Figure 26 – *War driving* dans la Silicon Valley**

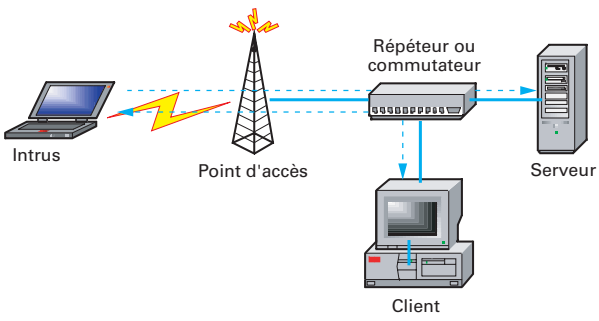


**Figure 27 – Attaque de l'homme au milieu en environnement sans fil**

strictement identiques (chiffrement WEP, canal...). Cela signifie donc qu'il est également très simple d'effectuer des attaques de l'homme au milieu, par exemple, en environnement sans fil.

La figure **27** montre que l'intrus a utilisé une des techniques que nous avons présentées (usurpation de l'adresse MAC, attaque de commutateur ou d'un protocole de routage, § 2.2.1) et a pu ainsi se placer en position de l'homme au milieu. Il peut ainsi se trouver au milieu :

— de plusieurs équipements sans fil reliés au même point d'accès ;

— d'équipements reliés sur un réseau traditionnel Ethernet filaire alors que lui-même n'est relié à ce réseau que par un point d'accès ;

— au milieu d'équipements reliés à des points d'accès différents, eux-mêmes reliés au même répéteur ou commutateur ;

— entre un équipement sans fil et un équipement relié à un réseau Ethernet filaire.

Il n'est plus nécessaire à l'intrus d'avoir un accès physique au réseau pour pouvoir l'attaquer.

■ **SSID :** avant tout, il faut clairement dire que le SSID (Service Set Identifier) n'est pas un mécanisme de sécurité. Ce n'est qu'un mécanisme qui vise à faciliter l'accès d'un client sur un équipement sans fil. Le SSID est en fait une chaîne de caractères libre (mais configurée avec une valeur par défaut) qui est diffusée sur les ondes radio pour annoncer la présence d'un point d'accès. Il suffit donc d'écouter le réseau pour découvrir l'existence d'un point d'accès et potentiellement connaître la marque et le modèle du point si le SSID est toujours celui par défaut.

Une mesure simpliste de sécurité consiste donc à configurer le point d'accès pour qu'il arrête d'envoyer des SSID. Dans ce cas, l'intrus qui veut se connecter au point d'accès doit prendre connaissance du SSID à appeler. Pour cela, le SSID n'étant jamais chiffré, il suffit à l'intrus d'écouter le réseau pour découvrir la demande de connexion d'un utilisateur et par la même occasion le SSID utilisé.

■ **Attaques WEP :** la principale faiblesse de la technologie sans fil se situe au niveau de WEP (Wired Equivalent Privacy). WEP est un protocole de sécurité défini dans le standard 802.11b, chargé d'assurer un niveau de sécurité équivalent à celui des réseaux filaires. En effet, le réseau filaire est par nature plus sécurisé que le réseau sans fil car la présence physique est nécessaire pour avoir accès au réseau. Les réseaux sans fil s'appuient sur des émissions radio et sont par conséquent plus faciles d'accès puisqu'il n'est plus nécessaire d'être physiquement présent dans l'infrastructure de l'entreprise dont on veut pénétrer le réseau. WEP a donc pour mission d'augmenter la sécurité en chiffrant les données transitant sur les ondes radio.

Pour assurer cette mission, WEP chiffre le paquet et le *checksum* de chaque trame 802.11 grâce à l'algorithme RC4, et les trames sont déchiffrées à l'arrivée chez le récepteur. Pour construire la clé qu'il utilise pour le chiffrement, WEP prépare une « graine » (*seed*) correspondant à la concaténation de la clé secrète fournie par l'émetteur et d'un vecteur d'initialisation (IV : Initialization Vector) généré aléatoirement sur 24 bits. Par ailleurs, un calcul d'intégrité (non chiffré) *via* un algorithme CRC32 est effectué sur les données. Ce calcul, appelé ICV (Integrity Check Value), est concaténé avec les données. La graine est ensuite utilisée dans un générateur de nombre aléatoire afin de générer la clé de chiffrement d'une longueur égale à la quantité de bits à transmettre. Le chiffrement des données se fait par un XOR (opération logique de OU exclusif) bit à bit entre cette clé et les données concaténées avec l'ICV, formant ainsi les données chiffrées. Elles sont transmises et l'IV est ajouté à la trame.

La clé secrète partagée peut être d'une longueur de 40 ou 64 bits, certaines versions offrant maintenant des clés allant jusqu'à 128 bits (en fait 104 bits). Mais même si la clé est plus longue et donc plus difficile à trouver, cela reste tout de même très insuffisant pour parler de véritable sécurité [H 5 210] [H 5 510].

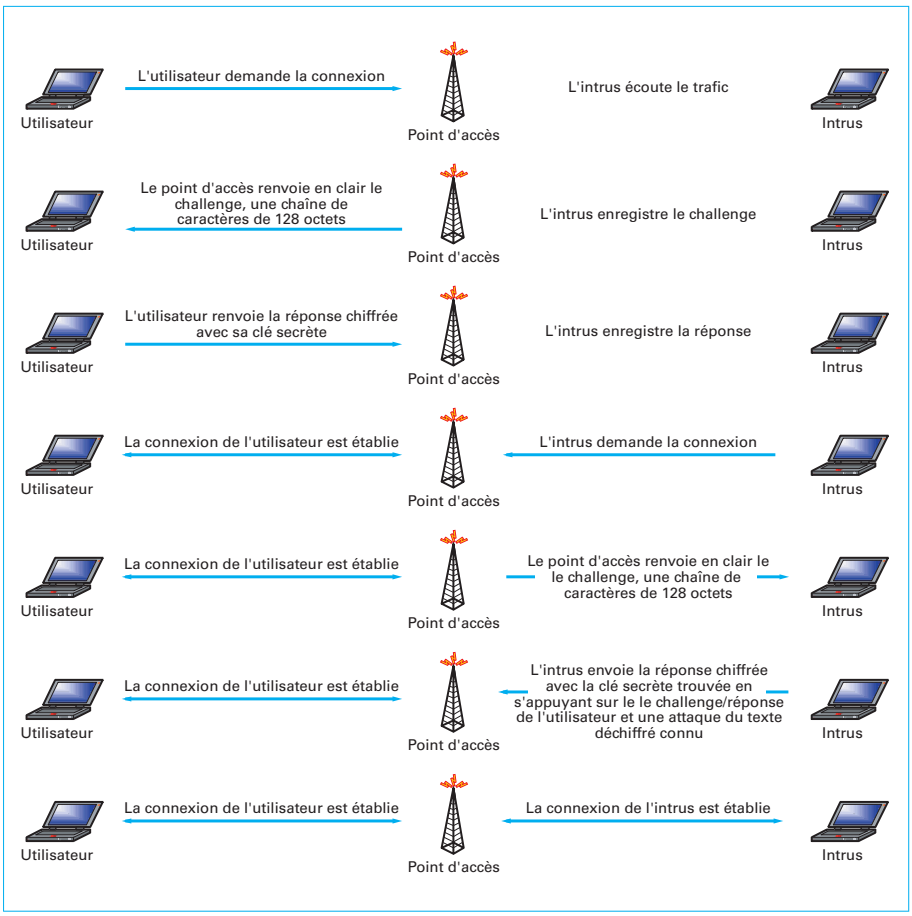


**Figure 28 – Attaque de la clé secrète sans utilisation de WEP**

● **Attaque FMS sur RC4 :** malheureusement, RC4 est connu depuis des années pour être vulnérable à des attaques de type « texte déchiffré connu » (*known plain text attack*). Ce type d'attaque consiste à deviner la clé de chiffrement en s'appuyant sur la connaissance de tout ou partie des données de la version déchiffrée. La technique d'attaque FMS (Fluhrer, Mantin et Shamir) a démontré qu'il fallait environ 20 000 paquets pour trouver la clé de chiffrement, ce qui représente 11 s de trafic.

Ainsi, lors d'une demande auprès d'un point d'accès avec une clé secrète et sur lequel WEP ne serait pas activé, la séquence des événements se déroulerait comme sur la figure **28**.

L'intrus a enregistré l'échange challenge/réponse de l'utilisateur et il sait que la réponse contient la version chiffrée avec la clé secrète du challenge. L'intrus connaît ce challenge puisqu'il a transité en clair lors de l'établissement de la session de l'utilisateur. L'intrus peut donc chercher la clé secrète en pratiquant une attaque de type « texte déchiffré connu » sur la réponse de l'utilisateur. Une fois que le challenge apparaît en clair dans le message réponse, la clé secrète est trouvée.

Si WEP est activé, le vecteur d'initialisation (IV) est le texte déchiffré connu à rechercher dans la réponse. En effet, l'IV reste non chiffré au sein des flux WEP. De plus, il faut savoir que WEP est connu pour produire des vecteurs d'initialisation très pauvres et avec une probabilité de répétition trop élevée. Pour cette raison, WEP autorise du trafic sur un vecteur obsolète, ce qui est utilisé dans le cadre d'une attaque.

Enfin, il existe d'autres méthodes pour trouver la valeur de la clé de chiffrement. Il suffit par exemple à l'intrus d'écouter le trafic jusqu'à ce qu'une collision de vecteurs d'initialisation se produise. Il lui suffit alors de passer les deux paquets ayant le même vecteur d'initialisation à travers une opération de OU exclusif pour obtenir la clé. Citons en dernier exemple une méthode où l'intrus génère tous les vecteurs d'initialisation possibles (il peut y en avoir $2^{24}$ dans une table). Ainsi, il peut tenter le déchiffrement en essayant chacun d'eux, ce qui est très rapide à réaliser avec les ordinateurs actuels.

● **Attaque par modification de paquet :** WEP utilise un *checksum* pour s'assurer de l'intégrité du paquet. Malheureusement, WEP utilise une fonction linéaire pour calculer ce *checksum*. À cause de cela, il est possible de modifier le contenu d'un paquet avec son *checksum* sans aucune détection de la part du récepteur. Cette attaque est également connue sous le nom de *bit flipping attack* car une variante consiste à simplement déplacer les bits.

● **Attaque par envoi de paquet ou par répétition :** nous avons vu précédemment qu'une partie de la clé repose sur le vecteur d'initialisation qui est généré aléatoirement. Malheureusement, il est possible de réutiliser un vecteur d'initialisation sans que cela soit considéré comme un comportement anormal, et donc tracé (*logged*), à cause de la pauvreté dans la génération de ces vecteurs.

Grâce à cette particularité, il est possible pour l'intrus, même lorsque l'échange entre le point d'accès et le client génère un nouveau vecteur d'initialisation pour chaque nouveau paquet de données, d'envoyer des paquets avec un vieux vecteur d'initialisation qui est déjà obsolète dans la communication entre le véritable client et le point d'accès. Cela permet aussi de pratiquer des attaques par répétition (*replay attack*).

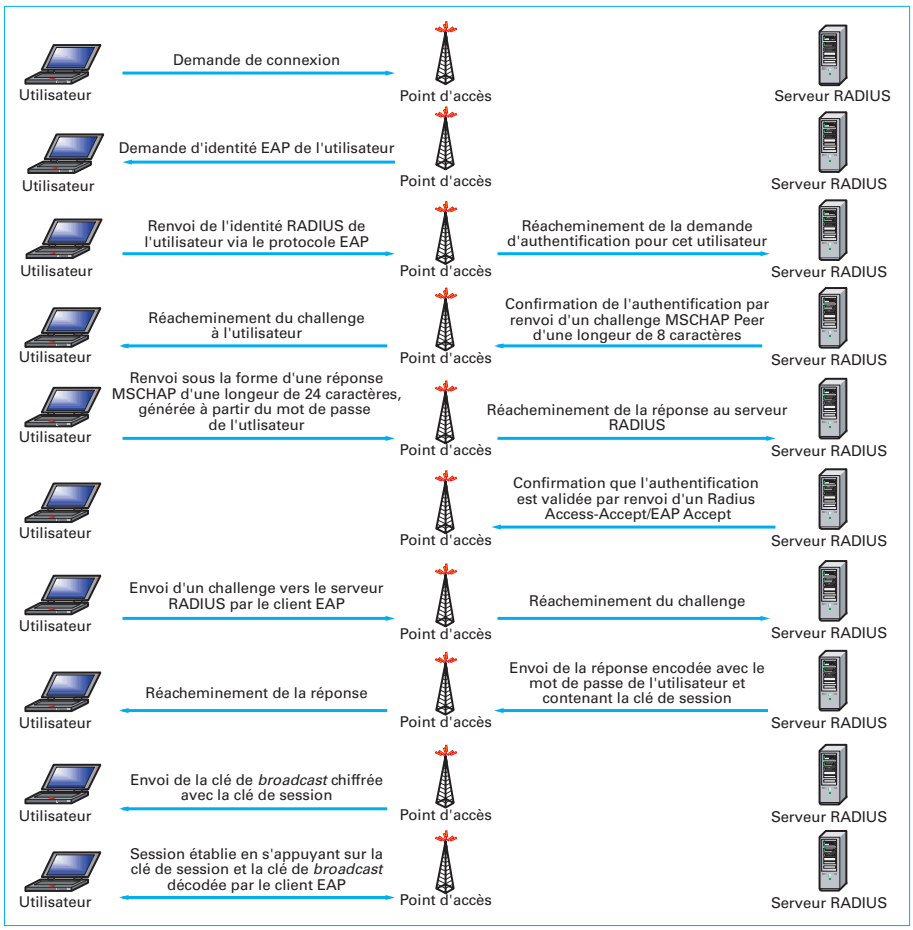


**Figure 29 – Authentification par le protocole LEAP**

● **Attaque par redirection d'adresse IP :** cette attaque nécessite que le point d'accès permette l'accès à Internet, ce qui est le plus souvent le cas, et que l'intrus contrôle un équipement sur Internet. La méthode est la suivante :

— l'intrus modifie l'intégrité d'un paquet en altérant l'adresse IP destinatrice du paquet en l'adresse de l'équipement qu'il contrôle. Il s'appuie pour cela sur un paquet capturé et la méthode du *bit flipping attack* ;

— il garde aussi une copie du paquet chiffré ;

— le paquet est donc déchiffré par le point d'accès puis envoyé en clair sur le réseau vers l'adresse IP destinatrice qui reçoit la version en clair du paquet de données ;

— l'intrus récupère alors cette version en clair.

L'intrus possède maintenant la version chiffrée et déchiffrée du paquet et il peut commencer une attaque de type « texte déchiffré connu » pour trouver la clé.

■ **Attaques LEAP :** LEAP (Lightweight EAP) est la réponse de Cisco pour pallier les faiblesses de WEP. LEAP repose sur le protocole 802.1X, et donc sur l'extension EAP (Extensible Authentication Protocol) de RADIUS (Remote Authentication Dial-In User Service) qui est utilisée pour l'authentification d'un client sur un équipement sans fil. Il faut donc construire une infrastructure s'appuyant sur RADIUS, et alors l'authentification se passe comme le décrit la figure **29**.

La clé de session est envoyée au client EAP (donc à l'utilisateur mobile) encodée à travers le mot de passe de celui-ci. Toute la sécurité de cette architecture repose donc sur la qualité des mots de passe qui est le plus souvent très faible, voire nulle.

Cela signifie surtout que LEAP est vulnérable à des attaques brutales de dictionnaire (*dictionary attack*) qui visent à trouver le mot de passe. Une fois celui-ci trouvé, toute la sécurité de la session tombe avec lui puisque la clé de session peut être connue, et par conséquent la clé de *broadcast* également.

En toute logique, quantités d'outils sont donc disponibles pour trouver ce mot de passe au travers d'une authentification LEAP. Ceux-ci sont par exemple : asleap, THC Leap, anwrap.

## 2.6 Attaques sur les protocoles de routage

Les réseaux reposent de nos jours sur des protocoles de routage de plus en plus intelligents. Ils ont pour mission de faire que le réseau soit capable de corriger par lui-même les chemins que doivent prendre les paquets, en détectant par exemple le meilleur chemin ou le déni de service d'un équipement particulier. Mais les pirates ont découvert des failles dans le raisonnement de ces protocoles, en plus des bogues de programmation qui sont apparus au fil du temps, et ont su les exploiter.

Ainsi, nous avons présenté (§ 2.4.3) l'attaque sur le protocole IRDP qui permet de modifier la route par défaut et autorise ainsi l'intrus à se placer en position de relais transparent. Nous abordons

ici des attaques sur de véritables protocoles de routage utilisant des algorithmes complexes.

### 2.6.1 Attaques génériques

Il existe, en plus des attaques spécialisées sur un protocole particulier, des techniques qui fonctionnent sur tous les protocoles. Nous pouvons déjà soupçonner qu'ARP permette d'engendrer de nouvelles attaques, mais nous allons découvrir ici bien d'autres possibilités de nuire à un réseau.

■ **Attaques par répétition :** les attaques par répétition (*replay attack*) sont basées sur le principe très simple où l'intrus a écouté le réseau et a pu enregistrer les échanges de données entre deux équipements. Il utilise ces échanges pour se faire passer pour un des équipements en les émettant à nouveau sur le réseau.

Les attaques de ce type contre un équipement de routage ont en général pour but d'engendrer un **déni de service**. L'intrus capture ainsi un échange de tables de routage et le « rejoue » sur le réseau plus tard, alors que le réseau a été modifié. Il provoque alors la remise en place des routes telles qu'elles étaient au moment de la capture de l'échange, ce qui change le routage en cours et peut donc provoquer un déni de service.

Mais l'intrus peut aussi réussir à modifier les informations véhiculées au sein des paquets rejoués, ce qui peut engendrer un changement de comportement du routage au sein du réseau.

■ **Technique de la fausse route :** elle est appelée également *false route attack*. L'intrus cherche à fabriquer des fausses routes. Dans ce cas, sa motivation peut être variable. Il peut désirer engendrer un déni de service en routant vers un équipement qui n'existe pas ou n'est pas chargé d'assurer du routage. Il peut aussi provoquer une situation où il se trouve en position d'écouter le réseau afin de capturer de l'information qu'il utilise pour gagner des privilèges ou pour lui permettre de faire des attaques par répétition (*replay*) par exemple. La variante de cette technique visant à engendrer une situation de l'homme au milieu est appelée attaque d'aiguillage (*shunt attack*).

### 2.6.2 Attaques OSPF

OSPF (Open Shortest Path First) est un protocole de routage faisant partie des protocoles IGP (Interior Gateway Protocol), ce qui signifie qu'il distribue les informations du routage entre routeurs appartenant au même système autonome (AS : *autonomous system*).

Un système autonome est un réseau ou un groupe de réseaux régi par une administration et des politiques de routage communes. Ainsi des réseaux d'utilisateurs finaux comme les entreprises privées emploient-ils généralement des protocoles IGP tels que RIP ou OSPF pour l'échange des informations de routage.

OSPF fonctionne sur TCP/IP, ce qui signifie que l'échange d'information s'appuie sur ce protocole. Dans OSPF, chaque routeur contient une base de données identique décrivant la topologie du système autonome. À partir de celle-ci, des tables de routage sont calculées afin de construire un arbre des chemins les plus courts (*shortest path*). OSPF peut calculer ses routes très rapidement lors d'un changement de la topologie tout en utilisant très peu de trafic réseau. La base de données AS décrit une représentation graphique de la topologie du réseau.

Lorsque aucune « aire OSPF » n'est configurée, chaque routeur du système autonome a une base de données identique débouchant sur la même représentation graphique. Un routeur génère ses tables de routage à partir de cette représentation en calculant l'arbre des chemins les plus courts en se considérant comme le routeur racine (*root*), ce qui signifie que le chemin le plus court dépend du routeur qui le calcule.

Le protocole Hello qui s'appuie sur un mode de diffusion est responsable de l'établissement et du maintien des relations de voisinage, un voisin étant un routeur situé sur la même interface. Il s'assure également que les échanges de données sont bidirectionnels et que les paquets sont envoyés régulièrement sur toutes les interfaces des routeurs. Dans les réseaux à accès multiples, le protocole Hello élit un DR (*designated router*) du réseau qui, entre autres tâches, est chargé de contrôler les interconnexions avec le réseau.

Le DR est responsable de la génération des annonces d'état des liens (LSP : Link State Packet ou LSA : Link State Advertisement) découlant de la collecte des accusés de réception de chaque LSP en provenance des autres routeurs. Compte tenu que le DR doit conserver énormément d'informations, cela demande beaucoup de temps et de trafic réseau de prendre le relais dans le cas où le DR vient à défaillir. C'est pour cela qu'OSPF choisit d'élire aussi un DR de sauvegarde (BDR : *backup* DR) qui contient les mêmes informations que le DR.

Suite à un manque de prise en compte de la sécurité dans le développement du protocole OSPF, les pirates ont découvert de multiples méthodes pour se faire passer pour un DR sur un réseau et ainsi être à même de modifier la totalité du routage de l'ensemble du réseau.

■ **Attaque du numéro de séquence maximal d'une annonce :** dans les spécifications d'OSPF v2, le champ réservé au numéro de séquence est un entier signé de 32 bits utilisé pour détecter les annonces anciennes ou dupliquées. Ainsi, lorsqu'un routeur reçoit un LSA, la valeur du numéro de séquence est comparée à celle de l'annonce actuelle afin de savoir lequel est le plus récent. Si les valeurs sont différentes, alors l'annonce avec le numéro de séquence le plus élevé est conservée.

Un routeur utilise donc 0x80000001 comme valeur de départ lorsqu'il envoie un LSA (la valeur 0x80000000 étant réservée), puis il incrémente le numéro de séquence d'une unité à chaque nouvelle annonce envoyée. La valeur maximale du numéro de séquence a été déterminée sur le postulat qu'un routeur n'est pas censé générer plus d'une annonce par 5 s (qui est l'intervalle de temps qui doit obligatoirement s'écouler entre deux émissions d'annonce) et qu'il lui faudrait donc $2 \times 2^{31}$ s (soit 340 ans) pour atteindre la valeur maximale d'un entier signé.

Par conséquent, lorsqu'une tentative est faite d'incrémenter le numéro de séquence au-delà de la valeur maximale (0x7FFFFFFF), l'annonce LSA courante doit en premier être ôtée du domaine de routage avant que la nouvelle annonce qui aurait un numéro de séquence égal à 0x80000001 soit envoyée. Si ce nettoyage n'est pas effectué, alors les autres routeurs considèrent le nouvel LSA comme plus vieux que l'existant car la première valeur au-delà du milieu d'un entier signé (0x80000001 donc) correspondant a une valeur négative.

En théorie, le LSA avec la valeur maximale doit être purgé du domaine de routage. Malheureusement, à cause de bogues d'implémentation, ce n'est pas le cas.

Un intrus qui envoie un LSA avec un numéro de séquence maximal fait que le routage est ajusté selon les informations fournies dans le LSA et que toutes les mises à jour suivantes sont ignorées par les routeurs.

■ **Attaque du numéro de séquence d'une annonce :** comme nous l'avons vu, le numéro de séquence OSPF est déterminant pour un routeur. Selon la valeur de ce numéro, le routeur considère en effet l'annonce comme plus récente (et donc valide).

Par conséquent, un intrus peut tout simplement envoyer une annonce avec un numéro de séquence supérieur à celui de l'annonce courante (qu'il peut déterminer simplement en écoutant le réseau). Le réseau ajuste alors son routage en fonction des informations incluses dans l'annonce.

### 2.6.3 Attaques BGP

Contrairement à OSPF qui est un protocole de routage fonctionnant au sein d'un système autonome (AS), BGP (Border Gateway Protocol) permet de fonctionner entre systèmes autonomes. Il est

donc utilisé par les fournisseurs d'accès à Internet (FAI) pour l'échange des informations de routage d'Internet.

Quand BGP est utilisé entre les systèmes autonomes, il est qualifié de BGP externe (eBGP : *external* BGP). À l'inverse, lorsque BGP est utilisé pour l'échange d'informations de routage au sein d'un AS, il est qualifié de BGP interne (iBGP : *internal* BGP).

BGP, qui en est à la version 4, est un protocole très robuste et plus adapté pour le travail à grande échelle. Il est donc capable de gérer les quelque 100 000 routes publiées sur Internet. Afin d'atteindre une telle performance, BGP s'appuie sur des paramètres de route appelés attributs, en plus du CIDR (Classless Inter Domain Routing) qui permet de s'affranchir de la contrainte des classes d'adresses et ainsi d'économiser les adresses utilisées. Toute cela lui permet de définir des politiques de routage et de garder stable l'environnement de routage.

Les voisins BGP (d'autres routeurs directement attachés sur la même interface et faisant aussi du BGP) s'échangent toutes les informations de routage au travers de la première session TCP établie. Quand des changements de routage sont détectés, les routeurs BGP envoient uniquement les différences entre les routages précédent et courant. Enfin, les routeurs BGP n'envoient pas régulièrement des mises à jour et les changements de routage ne font que modifier le chemin optimal vers un réseau final. Toutes ces informations sont suffisantes pour qu'une représentation graphique du réseau et de ses points d'interconnexion soit construite, d'où les boucles de routage peuvent être détectées et éliminées.

Un des problèmes engendrés par l'événement de l'AS 7007 (§ 3.2.5) était qu'un routeur publiait des routes qui ne lui appartenaient pas. Il est en effet possible d'envisager qu'un routeur indélicat puisse publier sur Internet des routes appartenant à d'autres entreprises ou fournisseurs d'accès, ou même des sous-réseaux qui n'appartiennent encore à personne. Il s'agit là d'un problème à part entière qui peut être mis à profit par un intrus.

Il faut comprendre la manière dont l'assignation d'adresses sur Internet, basée sur le RFC 2050, fonctionne. Elle définit une hiérarchie d'allocations partant du IANA (Internet Assignment Numbers Authority), puis viennent les RIR (Regional Internet Registries), puis agissant comme les LIR (Local Internet Registries) viennent les FAI, et enfin les clients de ces FAI en bout de chaîne. Dans ce système, personne n'est véritablement propriétaire des plans d'adressage mais chacun agit en tant que gestionnaire d'un sous-réseau dont la responsabilité lui est déléguée par son fournisseur.

Par conséquent, un client qui change de FAI perd toujours son plan d'adressage au profit d'un nouveau existant dans le sous-réseau délégué au nouvel FAI. Il est donc nécessaire de maintenir une base de données des sous-réseaux alloués, ce qui permet à un intrus de savoir qui est derrière chaque adresse IP présente sur Internet.

Ainsi, prenons le cas d'un routeur qui publierait des routes qui appartiennent à une entreprise privée. Si cette attaque se termine avec succès, alors cela peut engendrer deux situations :

— il y a bien sûr la probabilité d'un déni de service sur le véritable propriétaire du sous-réseau volé (puisqu'il s'agit finalement d'un vol). Dans ce cas, c'est l'intégralité du réseau volé qui est inaccessible depuis Internet, et non pas simplement quelques machines. Cela peut ainsi devenir une attaque de type « trou noir » ;

— cela peut permettre à un intrus de se placer en situation de l'homme au milieu. Il n'est en effet pas rare d'avoir des routeurs connectés entre plusieurs réseaux (pour du *peering*, par exemple, entre FAI). Ces routeurs sont particulièrement bien placés pour mener une attaque visant à créer une situation de l'homme au milieu car ils permettent de ne pas engendrer de perte du trafic détourné (contrairement à une attaque de type « trou noir »).

**Nota :** le *peering* est un accord que passent les FAI entre eux pour l'échange de données. Internet reposant sur des centaines de FAI, certains passent ces accords pour qu'il existe entre eux un lien réseau « équilibré ». On parle d'équilibre car aucun des deux FAI ne doit abuser l'autre en envoyant par exemple des données au point de *peering* qui ne devraient normalement pas passer par ce point. Ainsi, les données passent d'un FAI à l'autre plus rapidement puisqu'il existe une relation directe entre eux.

# 3. Engendrer un déni de service

Le pirate n'a pas toujours le désir de gagner des privilèges lorsqu'il attaque une entreprise. Il veut parfois simplement nuire en interrompant le fonctionnement d'une ressource comme un réseau ou un système. Pour atteindre ce but, il a recours à des techniques qui sont souvent bien plus simples à mettre en œuvre que celles que nous avons déjà vues.

## 3.1 Saturation de ressources

Parmi les méthodes les plus simples, citons la saturation des ressources du réseau (en général la bande passante) ou d'un système (nombre de sessions, bande passante...). Pour cela, il existe différentes techniques qui affectent tel ou tel élément de la ressource.

■ **Saturer le réseau par l'inondation :** le principe de l'inondation est très simple. Il se base sur le fait que le pirate peut générer plus de bande passante que la victime ne peut en offrir. La plupart du temps, le protocole utilisé pour inonder un système ou un réseau est ICMP.

Pourquoi ICMP ? Parce que, contrairement aux protocoles TCP et UDP par exemple, un équipement branché sur le réseau doit toujours analyser les paquets ICMP car ceux-ci demandent souvent une réponse particulière de sa part. Ainsi, un ICMP echo-request doit engendrer un ICMP echo-reply de même format (taille, numéro de paquet...), un ICMP Redirect doit être examiné pour en extraire les informations de routage, un ICMP Address Mask Request demande au récepteur de lui fournir son masque de sous-réseau, etc.

Afin de tester le comportement d'un lien saturé, de plus en plus de systèmes d'exploitation disposent d'une commande ping avec l'option d'inondation (encadré **3**).

Inonder une victime semble donc techniquement très simple, surtout si le pirate dispose de plus de bande passante que sa victime. Mais sinon, d'autres moyens existent.

■ **Saturer le nombre de sessions d'un système :** il existe une variante de l'inondation qui ne repose par sur le protocole ICMP et ne nécessite pas de disposer de plus de bande passante que la victime pour être efficace. Cette autre méthode a pour but de saturer les ressources de mémoire ainsi que d'atteindre les limites de la couche réseau du système d'exploitation cible.

Pour comprendre cette technique, il faut connaître le déroulement d'une session réseau au niveau de cette couche réseau réceptrice de la demande. La figure **30** montre comment le système se comporte dans la gestion des sessions du protocole TCP tant au niveau du client qu'au niveau du serveur. Sur cette figure, des annotations sont indiquées sur les flèches : XXX/YYY. Cela signifie que le paquet a envoyé les bits XXX activés, et que la réponse attendue en retour doit avoir les bits YYY activés.

Nous savons que le client est capable de fabriquer des paquets SYN sans suivre cet automate. Dans ce cas, les ressources allouées au niveau du client sont très rapidement libérées, puisqu'une fois le paquet construit, il est immédiatement envoyé par le client sans attente de réponse.

En revanche, nous constatons qu'au niveau du serveur, le processus de gestion est bien plus lourd et long. Une fois le paquet SYN reçu par le serveur, celui-ci alloue de la mémoire pour le stocker afin de le gérer et d'ajouter ce début de session dans sa table des sessions TCP, si bien sûr cette nouvelle session ne doit pas dépasser le nombre maximal de sessions autorisées. Puis, il renvoie un paquet SYN/ACK et attend que le client confirme le début de l'établissement de la session. Mais le pirate n'a pas le désir d'établir une session mais seulement de saturer le serveur et il ne répond donc pas.

**Encadré 3 – Extrait du manuel de la commande** ping
(d'après Free BSD)

```
NAME
   ping -- send ICMP ECHO_REQUEST packets to
network hosts
SYNOPSIS
   ping[-AQRadfnqrv] [-c count] [-i wait] [-l
preload][-m ttl]
   [-p pattern] [-P policy] [-s packetsize] [-S
src_addr] [-t timeout]
   [host | [-L] [-I interface] [-T ttl]
mcast-group]
DESCRIPTION
   Ping uses the ICMP protocol's mandatory
ECHO_REQUEST datagram to elicit an ICMP
ECHO_RESPONSE from a host or gateway. ECHO_REQUEST
datagrams (''pings'') have an IP and ICMP header,
followed by a ''struct timeval'' and then an
arbitrary number of ''pad'' bytes used to fill out
the packet. The options are as follows:
[...]
   -f Flood ping. Outputs packets as fast as they
      come back or one hundred times per second,
      whichever is more. For every ECHO_REQUEST sent
      a period ''.'' is printed, while for every
      ECHO_REPLY received a backspace is printed.
      This provides a rapid display of how many
      packets are being dropped. Only the super-user
      may use this option. This can be very hard on a
      network and should be used with caution.
[...]
```

Le serveur entre alors dans un état de fermeture des sessions partiellement ouvertes en envoyant au pirate un paquet FIN, puis en attendant deux fois la durée de vie du segment. Pendant cette période de temps qui prend plusieurs secondes et correspond aux états « Fin Attente-1 » et « Fin Attente-2 », le pirate a le temps d'envoyer des centaines (même sûrement des milliers) d'autres paquets SYN et finit par saturer les ressources de mémoire (allouée pour le réseau) et le nombre de sessions TCP du serveur. Par ailleurs, le pirate ne désirant pas établir de sessions avec le serveur, il peut usurper une ou plusieurs adresses IP afin qu'il soit impossible au serveur de localiser l'origine de l'attaque.

■ **Amplifier l'attaque** : le pirate peut cependant se trouver face à forte partie, un serveur particulièrement vigoureux placé sur un réseau relié à Internet avec une très forte bande passante. Dans ce cas, l'intrus, ne disposant pas des mêmes capacités, doit trouver un moyen d'amplifier son attaque.

Une des techniques d'amplification classique repose sur le *broadcast* déjà évoqué. Rappelons que toute machine recevant un paquet destiné à l'adresse de *broadcast* de son interface doit répondre à ce paquet.

Le pirate peut donc utiliser ce comportement pour saturer le réseau cible en envoyant simplement ses paquets vers l'adresse de *broadcast* du réseau cible. Toutes les machines visées répondront, surtout s'il s'agit d'un ping.

La figure **31** représente l'attaque :

— dans l'étape 1, l'intrus qui est connecté sur ADSL avec une capacité de 512 kbits/s diffuse un grand nombre de ping dans le réseau de sa victime ;

— dans l'étape 2, chacun de ces ping est reçu par les serveurs qui répondent. Cela engendre un trafic de 512 kbits/s de bande passante par serveur vers le commutateur ;

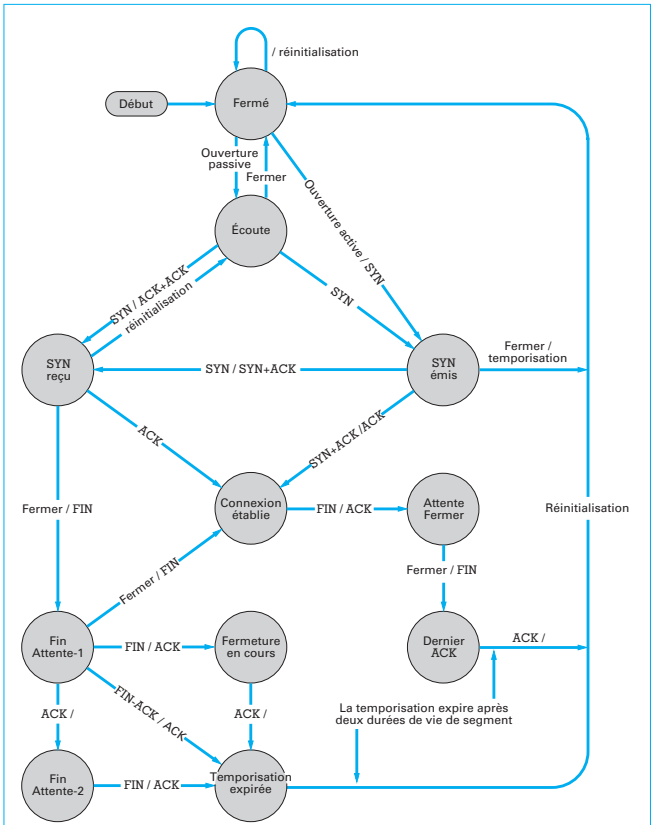


**Figure 30 – Automate à états finis de TCP**

— dans l'étape 3, le commutateur renvoie ses paquets vers Internet, d'où ils sont venus, engendrant alors un trafic de $5 \times 512 = 2{,}5$ Mbits/s de bande passante.

Bien sûr, l'intrus a pris soin d'usurper l'adresse source d'un serveur victime afin de ne pas être lui-même inondé de réponses.

Cette technique s'est améliorée par l'ajout du concept d'« architecture distribuée » ; ce qui est devenu par la suite une technique à part entière (§ 3.3).

## 3.2 Attaques génériques sur les protocoles de routage

Les protocoles de routage ont aussi des faiblesses qui, lorsqu'elles sont bien exploitées, permettent d'engendrer un déni de service.

### 3.2.1 Débordement de table d'un protocole de routage

Les protocoles de routage doivent maintenir dans une table la liste des routes qu'ils sont chargés de distribuer. La taille de cette table est limitée par la mémoire de l'équipement, mais également par la puissance de calcul qui est nécessaire pour extraire de la table l'information dont l'équipement a besoin pour décider de l'acheminement d'un paquet.

Une technique consiste à envoyer à un routeur un nombre de tables supérieur à ce qu'il est capable de supporter. Selon la marque et le modèle de routeur, les réactions diffèrent. Certains passent en déni de

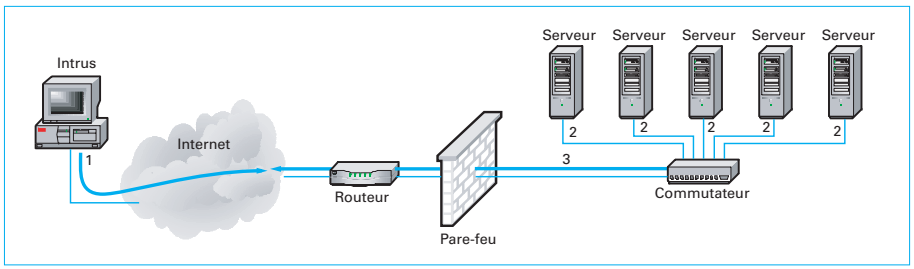


**Figure 31 –** ping ***broadcast*** **amplifié**

service car ils ont épuisé leur mémoire disponible, d'autres redémarrent, et enfin d'autres se contentent de ne plus pouvoir accepter de nouvelles routes, modifiant ainsi le comportement du réseau.

### 3.2.2 Usurpation d'adresse MAC ou IP

Nous avons vu précédemment le principe de l'usurpation de l'adresse MAC (§ 2.5) ou de l'adresse IP (§ 2.3). Cette technique peut être utilisée ici contre les équipements réseaux et non pas uniquement dans le but d'obtenir des privilèges.

Ainsi, un intrus peut envoyer des paquets de modification du routage en usurpant l'adresse MAC d'un ou de plusieurs équipements de réseaux, ce qui engendre des boucles (*loop*) du trafic et donc un déni de service. En effet, en temps normal, le paquet est envoyé à l'adresse MAC correspondant au récepteur du paquet ou au routeur chargé de router le réseau sur lequel est situé le récepteur. Si nous imaginons par exemple que le paquet est envoyé au mauvais routeur, celui-ci le renvoie vers sa route par défaut s'il en a une ou jette le paquet. Pour peu que la route par défaut pointe vers le routeur qui a envoyé le paquet, cela engendre une situation de boucle puisque les routeurs se renvoient ce paquet indéfiniment.

### 3.2.3 Technique du « trou noir »

Ce type d'attaque, qui s'appuie sur différentes méthodes selon le protocole de routage visé, a pour but d'engendrer un déni de service d'un réseau sous la forme d'une destruction des informations circulant sur le réseau. Pour être qualifiée de « trou noir » (*black hole attack* ou *sinkhole attack*), cette attaque doit provoquer deux situations :

— le routage doit être modifié afin que toutes les informations soient envoyées vers un équipement (en général un ordinateur) ;

— ledit équipement ne cherche pas à acheminer les données jusqu'à leur destination, mais se contente de les détruire.

Il ne s'agit donc pas d'une technique ressemblant à une attaque de l'homme au milieu puisque, ici, l'intrus ne cherche pas à usurper des privilèges en analysant les données d'une session, mais seulement à perdre l'information. Elle s'apparente davantage à la technique de la fausse route (§ 2.6.1).

Des équipements qui enverraient de fausses annonces pour faire croire qu'ils sont interconnectés entre tel ou tel réseau engendreraient des situations permettant à l'équipement de se placer en situation de l'homme au milieu s'il est vraiment un point de transit entre les réseaux auxquels il prétend être connecté. Dans le cas contraire, ils peuvent engendrer une situation de déni de service car l'interconnexion n'existe pas en réalité.

Par ailleurs, cette technique d'ajout de **routeurs fantômes** peut être également utilisée dans l'autre sens, à savoir le retrait de routeurs véritables. Enfin, l'envoi de telles annonces peut engendrer un comportement de *fight back* (§ 3.2.4).

Une variante consiste à envoyer des annonces dans lesquelles des liens inexistants ont été ajoutés.

### 3.2.4 Attaques contre OSPF

OSPF a lui aussi plusieurs faiblesses utilisables pour le placer en situation de déni de service.

L'âge maximal d'une annonce est d'une heure (3 600 s). Le principe de l'attaque consiste donc pour l'intrus à envoyer des annonces âgées de la valeur maximale possible.

Dans ce cas, le routeur propriétaire du LSA courant conteste ce changement par l'activation d'un processus dit de « contre-attaque » (*fight back*). Il renvoie alors l'annonce avec un âge égal à 0 et incrémente le numéro de séquence.

Une attaque qui exploite ce comportement par l'envoi continuel d'annonces trop âgées par exemple, engendre une inondation du réseau par les annonces envoyées/renvoyées avec les effets de bord (§ 3.2.5) comme l'incrémentation du numéro de séquence. De plus, ces trop nombreuses annonces finissent par corrompre la base de données contenant la représentation graphique du réseau à cause des mises à jour trop fréquentes. Tout cela peut mener finalement à une situation de déni de service.

### 3.2.5 Attaques contre BGP

Tout comme OSPF, BGP a lui aussi plusieurs faiblesses utilisables pour le placer en situation de déni de service.

■ **Interruption de sessions BGP :** parce que BGP utilise TCP et non pas UDP comme OSPF, il est bien plus difficile d'usurper des adresses IP, de détourner et de répéter une session, même si cela reste réalisable.

En revanche, TCP permet de disposer de nouvelles possibilités d'attaques. Ainsi, une session peut être interrompue par l'envoi depuis un intrus de paquets RST. Dans ce cas, une session TCP est relancée depuis le routeur cible ou le routeur avec qui il était en session, il n'y a donc techniquement pas de problème particulier. Mais si l'intrus persiste à envoyer de tels paquets, il peut engendrer des perturbations de routage, voire un déni de service, car lorsqu'une session est interrompue avec un voisin, les routes qui en proviennent sont toutes détruites.

■ **Fractionnement des routes :** le principe utilisé par BGP pour optimiser son comportement consiste à agréger les routes afin qu'elles fassent partie d'un seul sous-réseau lorsqu'elles doivent emprunter le même chemin, réduisant ainsi le nombre de routes à annoncer.

Un événement célèbre connu sous le nom de **AS 7007** se produisit sur Internet le 25 avril 1997. Ce jour-là, l'AS 7007, système autonome propriété de Florida Internet Exchange, ou FLIX, publia toutes les routes présentes sur Internet fractionnées en classes C (24 bits), engendrant ainsi l'ajout dans les tables du routage de dizaines de milliers de routes. De plus, l'AS 7007 annonça qu'il était propriétaire de la première classe C de chaque sous-réseau BGP existant, y compris ceux qui ne lui appartenaient pas. La plupart des routeurs avaient été « taillés » (quantité de mémoire, puissance de calcul...) pour pouvoir supporter le nombre de routes disponibles habituellement sur Internet et ce jour-là, ils furent à court de ressources, ce qui provoqua la perte de milliers de routes et une sacré pagaille...

Mais cet événement eu tout de même un effet positif car la communauté découvrit ce risque et pu ainsi développer une contre-mesure et améliorer la sécurité de ses configurations BGP.

Ainsi, alors qu'il était auparavant impossible de se prémunir contre le risque de recevoir des milliers de routes supplémentaires, BGP offre maintenant une limite du nombre maximum de routes qu'il peut accepter. Par ailleurs, cela a incité les NOC (Network Operation Center) à filtrer les routes qu'ils reçoivent de leurs voisins en n'acceptant que celles qui sont censées venir de ces routeurs. Malheureusement, si les FAI ont appris de cette crise, la plupart des entreprises qui ne sont pas des professionnels de la communication restent vulnérables à cette attaque.

■ **Instabilité des routes :** l'instabilité des routes, *route flapping* ou *BGP flapping*, est une situation où une route change constamment, ce qui contraint le routeur à recalculer ses tables et sature ses ressources.

Dans le milieu des années 1990, le trop grand nombre d'instabilités provoquait des problèmes de routage sur Internet. Pour lutter contre ce problème, un nouvel algorithme proposé par Curtis Villamizar a été mis en place (RFC 2439) dont le principe est simplement d'éliminer les routes instables.

Mais cette solution, qui résolvait ces problèmes, présentait également des faiblesses que les intrus ont su trouver. Ce qui était au départ un progrès s'est donc transformé en la possibilité d'engendrer un déni de service. Il suffit en effet à l'intrus de rendre instable une route pour que le routeur la détruise simplement de ses tables de routage. Heureusement, le RIPE-NCC publia une recommandation connue sous le nom de *RIPE-229 – RIPE Routing-WG Recommendations for Coordinated Route-flap Damping Parameters* qu'il suffit d'appliquer pour réduire le risque associé à cette attaque.

■ **Injection de routes DUSA :** certains sous-réseaux ont un usage réservé et ne devraient jamais être publiés sur Internet. Ces réseaux appelés DUSA (Designated Special Use Address) et définis par l'IANA sont :

- 0.0.0.0/8 route par défaut ;
- 0.0.0.0/32 *broadcast* ;
- 127.0.0.0/8 *loopback* ;
- 192.0.2.0/24 sous-réseau destiné à usage de test et documentation ;
- 10.0.0.0/8 adresses privées définies dans le RFC 1918 ;
- 172.16.0.0/12 adresses privées définies dans le RFC 1918 ;
- 192.168.0.0/12 adresses privées définies dans le RFC 1918 ;
- 169.254.0.0/16 sous-réseau en cas d'échec d'assignation DHCP ;
- 192.88.99.0/24 RFC 3068 préfixe *anycast* pour les routeurs relais IPv6 à IPv4.

Ces routes ne sont donc censées exister qu'au sein d'un réseau d'entreprise et un intrus qui réussirait à faire publier ces routes pourrait engendrer des perturbations sur les réseaux qui les utilisent.

### 3.2.6 Attaques *wireless*

Comme dans les réseaux filaires, il est possible d'engendrer des attaques de déni de service contre un point d'accès ou entre des équipements d'un environnement sans fil. Certaines de ces attaques existent également dans l'environnement filaire (par exemple, celles basées sur l'adresse MAC), mais d'autres sont spécifiques à l'environnement sans fil.

■ **Attaques de l'accès au réseau :** dans un environnement sans fil, la détection des collisions de paquets n'est pas chose aisée. Afin de pallier cette difficulté, des mécanismes basés sur la détection de porteuse virtuelle ou physique combinés à un contrôle d'accès sont utilisés. Ces mécanismes peuvent être mis en œuvre par un intrus.

Il existe quatre fenêtres de temps destinées à rendre l'accès au média prioritaire. Avant d'envoyer un message, une de ces quatre fenêtres de temps doit être respectée. Parmi elles, deux nous intéressent particulièrement. Il s'agit de SIFS (Short Interframe Space) qui est utilisé pour la continuation d'un échange déjà initié et DIFS (Distributed Coordination Function Interframe Space) qui est utilisé pour l'initialisation d'un nouvel échange. Afin d'éviter que tous les nœuds émettent dès la fin de l'expiration du DIFS, le temps qui suit le DIFS est divisé en portions. Chacun choisit aléatoirement la portion qu'il utilise pour commencer à transmettre. En cas de collision, l'émetteur génère aléatoirement un délai d'attente avant de recommencer à émettre.

Puisque chaque nœud doit systématiquement attendre au moins un délai SIFS (qui est utilisé après que la session a été initialisée en lieu et place du DIFS), il est possible qu'un intrus accapare tout le canal par l'envoi d'un signal juste avant la fin de l'expiration d'un SIFS. Bien sûr, cette attaque est fortement consommatrice en énergie puisqu'un SIFS est égal à 20 µs, ce qui signifie qu'il faut envoyer 50 000 messages par seconde pour que l'attaque soit efficace.

Mais il existe une attaque encore plus efficace s'appuyant également sur les mécanismes de détection de porteuse virtuelle utilisés pour éviter les collisions. Chaque message 802.11 inclut un champ qui précise le nombre de microsecondes pendant lesquelles le canal est réservé. Cette valeur est utilisée par le vecteur d'allocation du réseau (NAV : Network Allocation Vector) de chaque nœud, et c'est seulement lorsque cette valeur est égale à 0 qu'il est à nouveau possible de transmettre. Ce procédé est basé sur le mécanisme RTS/CTS (Request To Send/Clear To Send) utilisé pour synchroniser l'accès au canal.

Durant un RTS/CTS, le nœud émetteur envoie un message RTS indiquant une durée de réservation suffisamment importante pour que la séquence RTS/CTS complète puisse être effectuée. Le nœud récepteur répond au message RTS par un CTS incluant une valeur de durée égale à celle indiquée dans le RTS moins le temps écoulé à l'émission du CTS. Chaque nœud du réseau met à jour son NAV afin de différer l'accès au média jusqu'à la fin de la séquence RTS/CTS. Notons que ce mécanisme du RTS/CTS est peu utilisé en 802.11.

Compte tenu que n'importe quel message inclut cet indicateur de durée, un intrus qui émet avec la valeur maximale de durée de réservation du média (égale à 32 767 µs en 802.11b), 30 messages par seconde, monopolise toute la bande passante pour ses communications au détriment de celles des autres nœuds du réseau, engendrant ainsi un déni de service pour ceux-ci. Bien sûr, ces messages, comme tous ceux sur lesquels nous nous sommes appuyés pour les dénis de service précédents, ne s'appuient pas sur les mécanismes de chiffrement ou d'authentification, rendant ainsi ces techniques d'attaque encore plus aisées.

■ **Attaque de désauthentification :** nous avons vu dans les faiblesses du protocole WEP (§ 2.5) que l'utilisateur s'authentifiait auprès du point d'accès pour pouvoir utiliser celui-ci et accéder au réseau. Le protocole WEP inclue également un mécanisme visant à annuler cette authentification. Ce mécanisme avait été créé au départ dans le but d'éliminer le risque qu'un intrus prenne la suite d'une session légitime d'un utilisateur qui aurait cessé de communiquer.

Malheureusement, suite aux nombreuses faiblesses de WEP, nous savons maintenant qu'un intrus peut décoder/modifier des paquets WEP et se faire passer pour l'utilisateur connecté (voir les attaques de l'homme au milieu). L'intrus a donc déjà la possibilité de nuire fortement.

Le comble dans cette attaque est que le message de « désauthentification » ne s'appuie sur aucun mécanisme d'authentification ou de chiffrement. Par conséquent, il peut aisément être usurpé étant donné qu'il n'est pas nécessaire de casser le chiffrement WEP pour exécuter l'attaque. L'intrus peut donc envoyer un message demandant à ce que l'utilisateur sans fil soit « désauthentifié » afin que sa session s'arrête. Ce message peut être envoyé indifféremment au point d'accès ou à l'utilisateur. L'équipement visé change alors de comportement et ignore systématiquement tout message qui arrive de l'équipement « désauthentifié » qui n'est pas une nouvelle demande d'authentification. Cela engendre donc un déni de service pour l'utilisateur. Si l'intrus émet ces messages de « désauthentification » après chaque nouvelle authentification réussie, l'utilisateur ne peut plus travailler avec le point d'accès.

Par ailleurs, il est important de noter que cette attaque, pour être efficace, nécessite d'écouter tous les canaux d'émission/réception

(14 canaux en tout sur différentes fréquences) afin de détecter toutes les demandes d'authentification qui sont envoyées vers d'autres points d'accès. L'utilisateur peut en effet penser que son point d'accès a un problème et échapper à cette situation en changeant de point d'accès.

■ **Attaque de dissociation :** il est possible pour un utilisateur sans fil de s'authentifier sur plusieurs points d'accès en même temps. Dans cette situation, le standard 802.11 a prévu un message particulier d'« association » visant à définir quel point d'accès a la responsabilité de réacheminer les messages vers le réseau filaire pour le compte de l'utilisateur.

Malheureusement, tout comme pour le message de « désauthentification », le message de dissociation ne s'appuie pas sur le chiffrement et l'authentification de l'utilisateur. Il peut donc être facilement usurpé. Un intrus peut donc envoyer ce type de message pour engendrer un déni de service. Notons que cette attaque est moins efficace que l'attaque de « désauthentification » car le processus d'association est bien moins lourd que le processus d'authentification.

■ **Attaques basées sur l'économie d'énergie :** dans le but d'économiser l'énergie (car on présume que les ordinateurs clients sans fil sont majoritairement des portables fonctionnant sur batterie), le standard 802.11 a prévu un mécanisme où le client demande au point d'accès de ne plus envoyer de données pendant un laps de temps donné. Dans ce cas, le point d'accès stocke les données qu'il destine au client dans une mémoire tampon. Par la suite, le client demande périodiquement au point d'accès s'il a des données pour lui et, dans ce cas, de les lui envoyer, ce qui vide la mémoire tampon. Malheureusement, les messages qui gèrent ce mécanisme ne s'appuient toujours pas sur le chiffrement et l'authentification de l'utilisateur. Un intrus peut donc aisément les usurper. En conséquent, il est possible pour un intrus :

— d'engendrer des perturbations dans les communications entre le client et le point d'accès, en usurpant un message de demande de mise en sommeil (et donc du stockage des données dans la mémoire tampon du point d'accès) ;

— d'éliminer des données stockées sur le point d'accès en envoyant un message usurpé du client afin que le point d'accès envoie ces données alors que le vrai client est en sommeil et donc ne les recevra pas ;

— d'usurper ces messages, la présence de données en attente étant indiquée par le point d'accès sous la forme de *broadcasts* TIM (Traffic Indication Map), ce qui permet de faire croire au client que le point d'accès n'a aucune donnée à lui envoyer. Celui-ci retourne alors en mode sommeil ;

— de désynchroniser client et point d'accès : les messages TM ne s'appuient pas sur des mécanismes d'authentification et de chiffrement. Ils contiennent des informations permettant au client et au point d'accès de se synchroniser afin que le client sache quand il doit être à l'écoute du réseau. Un intrus envoyant des paquets usurpés engendre une désynchronisation du client et du point d'accès, et donc ils ne communiquent plus.

## 3.3 Déni de service distribué

Nous avons abordé le sujet du déni de service. Nous savons donc qu'une des volontés d'un intrus est de rendre inopérant le réseau ou la plate-forme de la victime. Nous avons également vu qu'il était possible d'amplifier une attaque (§ 3.1). Cependant, ce type d'attaque est devenu très difficile à mener car la plupart des réseaux ont pris les mesures appropriées pour ne plus en être victimes.

Les pirates ont alors étendu cette technique en la faisant reposer sur un outil directement créé par eux et qui se comporte en agent. Le DDoS (Distributed Denial of Service) est ainsi né.

Le principe du DDoS consiste donc à s'appuyer sur des réseaux ou plates-formes présents sur Internet afin de disposer d'un maximum de ressources (en général de la bande passante ou de la capacité à générer des sessions) à exploiter face à une victime.

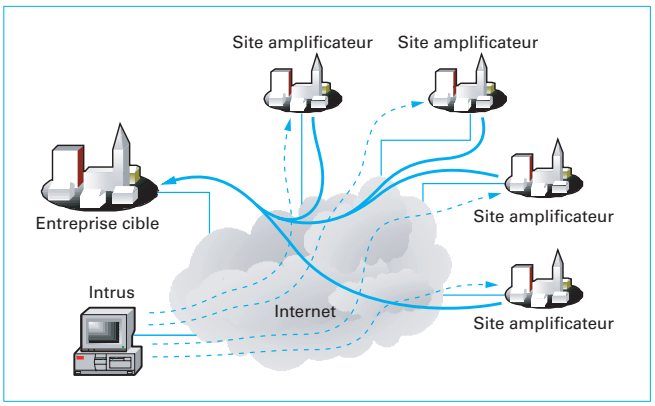


**Figure 32 – Amplification distribuée**

### 3.3.1 *Smurf* et *fraggle*

L'utilisation d'outils n'est pas toujours nécessaire et il existe une forme distribuée de l'inondation amplifiée appelée *smurf*. Elle consiste à utiliser des réseaux vulnérables à l'attaque du ping *broadcast*, pourvu qu'ils soient particulièrement amplificateurs. L'intrus utilise donc dans son inondation de ping *broadcast* des réseaux énormément fournis en machines répondant au ping *broadcast*.

Sur la figure **32**, l'intrus inonde de ping *broadcast* des sites amplificateurs (trait en pointillé). Ce paquet echo-request usurpe l'adresse *broadcast* du réseau victime de l'attaque finale. Les sites amplificateurs répondent en amplifiant l'attaque, mais à l'adresse source, elle-même une adresse *broadcast* du réseau cible. Le réseau cible reçoit cette attaque déjà amplifiée et l'amplifie peut-être même à son tour. Nous pouvons facilement imaginer un scénario où l'intrus inonde les sites amplificateurs à raison de 512 kbits/s chacun. Chaque site amplifie 20 fois par exemple, ce qui engendre un trafic retour de $4 \times 20 \times 512 \approx 40$ Mbits/s de bande passante générée par 80 machines. Ce trafic arrive sur le réseau finalement victime de l'attaque qui doit être capable de disposer de plus de 40 Mbits/s de bande passante pour répondre à ces clients légitimes. Par ailleurs, si un pare-feu est situé dans le chemin vers le réseau victime, un déni de service est possible si tous ces paquets sont tracés (*logs*), le système devant écrire sur ses disques à une cadence infernale.

Il existe une version légèrement différente de cette attaque appelée *fraggle*. Dans cette version, l'intrus envoie des paquets UDP (vers les services echo ou chargen en général) au lieu d'envoyer des paquets ICMP (ping).

Heureusement, ce type d'attaque ayant été largement utilisé sur Internet, la plupart des fournisseurs d'accès et des entreprises ont maintenant pris l'habitude de ne pas laisser passer les paquets destinés à des adresses *broadcast*.

### 3.3.2 Outils de DDoS

Contrairement à un général qui dispose d'un état-major s'appuyant sur des échelons de commandement qui pilotent les troupes, l'intrus doit se débrouiller seul. Il a alors imaginé le concept d'« agent dormant » (un peu à la manière des services secrets pendant la Guerre froide).

Ainsi que l'illustre la figure **33**, l'architecture DDoS consiste en plusieurs niveaux :

— le commandant en chef est la machine de l'intrus. C'est elle qui donne l'ordre d'attaque ;

— le premier niveau de commandement qui reçoit cet ordre est appelé *handler*. Il s'agit des machines qui relaient aux troupes d'ordre d'attaquer ;

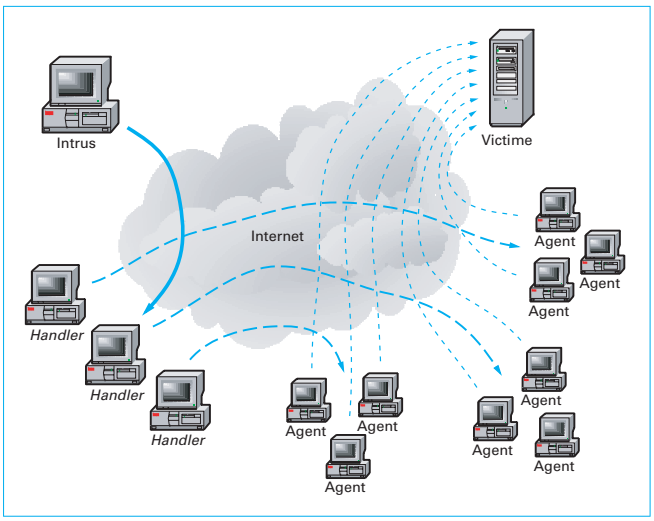


**Figure 33 – Architecture DDoS**

— la troupe est en fait un grand nombre d'agents dont la fonction unique est d'exécuter l'ordre reçu par les *handlers* ;
— enfin, la victime subit le résultat de ce schéma d'attaque.

Bien sûr, chaque *handler*, chaque agent a également pour fonction de se propager sur des machines victimes afin que la troupe et le premier niveau de commandement soient aussi nombreux que possible.

Les termes *handler* et agent sont ceux retenus par le CERT (Computer Emergency Response Team), organisme qui envoie régulièrement des alertes concernant les vulnérabilités des systèmes d'exploitations et applications, ainsi que sur les attaques majeures en cours (DDoS, ver...).

Historiquement, TFN (Tribe Flood Network) a été la première attaque DDoS, puis sont venus Trinoo et Stacheldraht.

■ **TFN** a empoisonné la vie des internautes durant l'année 1999. Il s'agissait d'un programme binaire fonctionnant sous Solaris Sparc 2.x et utilisant des vulnérabilités de débordement de pile présentes dans les services réseaux rpc.statd, rpc.cmsd et tpc.ttdbserverd. Le *handler* (tribe.c) était appelé client et l'agent (td.c) était appelé démon, l'intrus contrôlant un ou plusieurs clients, eux-mêmes contrôlant d'innombrables démons.

Le contrôle à distance d'un réseau TFN se faisait par des lignes de commande envoyées par des interpréteurs de commande accessibles par un port TCP, UDP ou ICMP tel que LOKI, SSH, telnet... Aucun mot de passe n'était nécessaire pour ordonner au client, il suffisait simplement de disposer de la liste de ceux-ci dans un fichier iplist.

Le client TFN étant contrôlable *via* ICMP, le seul moyen efficace d'empêcher son fonctionnement était de bloquer les paquets ICMP quelle que soit leur provenance.

■ **Trinoo** (appelé également Trin00) utilisait les mêmes faiblesses que TFN. En revanche, Trinoo était également disponible sur des plates-formes Linux.

Au niveau de l'attaque, Trinoo s'appuyait sur UDP. Au niveau de la communication, il utilisait différents ports et protocoles. Ainsi, entre l'intrus et le maître (ou *handler*), le port TCP 27665 était utilisé. Entre le maître et le démon (ou client), le port 27444 UDP était utilisé, et enfin entre le démon et l'intrus, le port UDP 13335. Le maître s'appuyait sur un programme master.c et le démon sur un programme ns.c.

Mais pourquoi utiliser des ports TCP et UDP et non pas un seul protocole ? La réponse est simple. Une session TCP requiert une remise fiable et un échange de données. Par conséquent, un réseau à la limite de la saturation ne peut pas communiquer efficacement en TCP. En UDP, c'est différent, car le serveur peut recevoir un ordre sans avoir à donner une réponse. Ainsi, il devient plus facile de communiquer au travers de réseaux dont la bande passante est saturée. C'est d'ailleurs la raison pour laquelle UDP est souvent utilisé pour faire de l'inondation SYN par les outils de DDoS.

Une autre différence entre TFN et Trinoo est qu'il utilisait une authentification par un mot de passe. Ceux-ci étaient par défaut :

— 144adsl pour le démon ;
— gOrave au démarrage du serveur maître ;
— betaalmostdone pour l'interface du maître ;
— killme pour la commande mdie qui tuait Trinoo.

L'interface fournissait un ensemble de commandes permettant de contrôler le démon, spécifier les cibles, etc.

■ **Stacheldraht** (fil de fer barbelé) est un assemblage des fonctionnalités de TFN et de Trinoo. Il s'appuyait sur un programme *handler* (mserv.c) et sur des agents (leaf/td.c). La communication avec les *handlers* se faisait au travers d'un programme d'aspect telnet (telnetc/client.c). Cette communication s'appuyait sur le port TCP 16660 entre l'intrus et les *handlers* et le port TCP 65000 ou l'ICMP echo-reply pour la communication entre le *handler* et les agents. Enfin, l'authentification entre l'intrus et le *handler* se basait sur une clé de chiffrement symétrique.

Au niveau des fonctionnalités, Stacheldraht fournissait des commandes supplémentaires pour connaître le nombre d'agents considérés comme actifs ou morts, mais également afin de permettre une meilleure maîtrise de l'architecture par des commandes mettant à jour les listes d'agents et de *handlers*. Enfin, l'interface permettait également de modifier des paramètres du réseau tels que la taille des paquets ICMP ou UDP, l'intervalle des ports réseaux sources, etc.

### 3.3.3 Le pire ennemi : le ver

Le premier programme qui a eu le comportement d'un ver a été créé par Bob Thomas en 1971. Il s'agissait d'un programme écrit pour satisfaire un besoin des contrôleurs aériens dont le but était de notifier aux opérateurs que le contrôle d'un vol particulier était passé d'un ordinateur à un autre. Ce programme, appelé *creeper*, n'a alors réussi qu'à passer d'un système à l'autre en affichant sur l'écran *I'm creeper ! Catch me if you can* ! (Je suis creeper ! Attrapez- moi si vous le pouvez !) Ce programme ne se reproduisait pas.

Ce n'est véritablement qu'à partir des années 1980 que des chercheurs de Xerox créèrent cinq programmes de type ver dans le but de rendre différents services aux utilisateurs d'un réseau en indiquant par exemple qu'un message venait d'être posté. Le nom *worm* qui provient du roman *The Shockwave Rider* de John Brunner (1975) a alors été explicitement utilisé.

Puis à nouveau le concept de ver fut oublié jusqu'en 1988 où un jeune étudiant nommé Robert Morris de l'université de Cornell créa le premier ver dont l'existence a été néfaste puisqu'il a engendré quantité de dénis de service sur Internet, contraignant des équipes prestigieuses telles que celles du MIT (Massachusetts Institute of Technology) et de l'université de Berkeley à récupérer le programme afin de le désassembler et de comprendre comment une telle chose avait pu se produire.

Voici brièvement ce qui se passa le 2 novembre 1988 :

— à 18:00 environ, le ver fut lancé sur Internet ;
— à 20:49, il infecta un Vax 8600 de l'université de l'Utah ;
— à 21:09, il commença ses premières attaques visant à infecter d'autres systèmes depuis le Vax ;
— à 21:21, la charge du Vax était de 5 (à cette heure-là, la charge était habituellement autour de 1), sachant qu'au-delà de cette valeur, le système commence vraiment à avoir du mal à calculer ;
— à 21:41, la charge du Vax était de 7 ;
— à 22:01, la charge du Vax atteignait 16 ;

— à 22:06, la machine était si infectée qu'il n'était plus possible de lancer de nouveaux processus ;
— à 22:20, l'administrateur tua tous les vers présents sur le Vax ;
— à 22:41, le système fut réinfecté et la charge atteignit alors 27 ;
— à 23:21, l'infection engendrait une charge de 37.

En résumé, le système a été en situation de déni de service environ 90 min après avoir été infecté. Ce scénario s'est reproduit sur plus de 6 000 systèmes à travers tous les États-Unis et engendra des pertes financières estimées de 100 000 à 10 000 000 $.

**Définition :** un programme est considéré comme un ver s'il a un ensemble de comportements bien particuliers [H 5 440]. La définition stricte du ver est :

> Programme autonome et parasite, capable de se reproduire par lui-même, en perpétuel déplacement dans la mémoire de l'ordinateur qu'il surcharge et mine progressivement, consommant jusqu'à la paralysie les ressources du système informatique.

Cette définition n'est pas tout à fait exacte. Le ver est bien un programme autonome dont la mission principale est de se dupliquer sur un aussi grand nombre de systèmes informatiques que possible. Cependant, il ne cherche pas à surcharger le système qui l'héberge car ce serait se suicider. En fait, il utilise les ressources du système infecté pour se dupliquer, ce qui engendre une charge de travail supplémentaire. Contrairement au virus informatique, le ver ne cherche pas à détruire des informations présentes sur le système. En revanche, il n'est pas rare que le ver dépose un programme « cheval de Troie » qui permet de prendre le contrôle de la machine à l'insu de son utilisateur. Dans la plupart des cas, le ver s'appuie sur une vulnérabilité pour se dupliquer. Cela lui garantit une meilleure chance d'infecter le système cible. Par ailleurs, le ver ne cible pas ses victimes. Dans le meilleur des cas, il utilise des informations présentes sur la machine infectée (adresse IP, masque de sous-réseau, carnet d'adresses mail...) pour être plus efficace, mais le ciblage ne va pas plus loin.

Enfin, il arrive que le ver sature des ressources du système infecté ou du réseau sur lequel il est connecté. Cela vient soit d'un algorithme agressif de duplication, soit d'une utilisation du système comme agent pour mener une attaque de déni de service distribué vers une victime définie (Yahoo, Microsoft, etc.).

**Les nouveaux vers :** si on analyse le premier ver avec un peu de recul, on constate qu'il n'a finalement pas été si méchant et néfaste. De nos jours, les dommages sont toujours aussi vastes alors que la taille d'Internet a été multipliée plus d'un million de fois.

Pourquoi tant de dommages ? Parce que le ver se retrouve sur des systèmes de particuliers qui ne sont pas conscients qu'ils sont infectés et qui souvent ne savent pas quoi faire pour résoudre cette situation. Par conséquent, les vers restent présents pendant des mois sur des systèmes qui sont en plus éteints quand ils ne sont pas utilisés, ce qui rend encore plus difficile leur détection.

De plus, grâce à l'avènement de l'informatique de l'Internet grand public, le nombre de systèmes informatiques peu sécurisés et mis à jour a également fortement grossi, ce qui laisse de vastes possibilités d'infection aux vers.

Enfin, il faut clairement noter que le système d'exploitation le plus utilisé au monde reste Windows de Microsoft pour lequel on découvre chaque semaine une nouvelle faille pouvant être utilisée par un ver.

De nos jours, les vers infectent les autres systèmes non seulement *via* le réseau en exploitant telle ou telle vulnérabilité permettant d'exécuter un programme arbitraire sur la victime, mais ils utilisent des techniques d'ingénierie sociale en exploitant les carnets d'adresses des victimes et leur logiciel de messagerie, envoyant des messages en remplaçant l'adresse source par une autre, usurpée du carnet d'adresses.

# 4. Conclusion

Il existe quantités de faiblesses pouvant être exploitées par une personne mal intentionnée désireuse d'accéder sans autorisation à des informations privées, ou d'empêcher un réseau informatique, qu'il soit filaire ou sans fil, de fonctionner correctement. Par ailleurs, toutes ces possibilités de nuire qui ont été énoncées ne se sont jamais appuyées sur le principe d'une intrusion de l'équipement réseau, qui est l'ultime moyen pour un pirate de faire faire ce qu'il veut au réseau, mais uniquement sur les faiblesses du protocole TCP/IP et des services réseaux assurant le fonctionnement correct de celui-ci. Enfin, nous n'avons pas montré comment certaines de ces techniques utilisées ensemble se complétaient afin d'augmenter considérablement leur pouvoir de nuisance. Comme on peut le constater, les vers les plus récents, qui en une seule infection empêchent l'antivirus de se mettre à jour en faisant pointer tous les noms de machines des vendeurs d'antivirus vers 127.0.0.1 (*localhost*), installent un cheval de Troie et continuent à infecter les systèmes attachés à un réseau par exemple.

Mais comme l'introduction l'annonçait clairement, toutes ces faiblesses découlent souvent d'un manque de « conscience de sécurité » que toute personne travaillant dans l'informatique devrait avoir, depuis le plus simple utilisateur jusqu'au directeur en passant par les développeurs, administrateurs et opérateurs. Si cette conscience existait, les équipements réseaux, les systèmes s'appuyant sur ces réseaux ainsi que les services réseaux seraient correctement configurés et sécurisés. Bien sûr, cette pratique a un prix. Il s'agit en général de contraintes supplémentaires lors d'installations, donc de temps supplémentaire passé, que ce soit pour sécuriser un nouvel équipement ou pour mettre en œuvre manuellement l'ajout d'un système, les automatismes qui facilitent la vie de chacun étant désactivés car source de faiblesses majeures.

Mais il ne faut pas non plus sombrer dans la paranoïa. Si une entreprise n'est pas reliée à un réseau qui n'est pas sous son contrôle comme Internet ou celui d'une tierce partie, si de plus elle n'utilise qu'un réseau filaire courant au travers d'une infrastructure qui lui appartient totalement et qui est sous son contrôle physique complet, alors cette entreprise ne risque pas grand-chose de personnes situées hors de l'entreprise elle-même. Rappelons à ce propos que 80 % des incidents informatiques avec une volonté de nuire (par opposition à un virus par exemple), proviennent du personnel de l'entreprise. Il faut donc bien relativiser le risque.

Enfin, il faut rappeler qu'il existe d'autres moyens pour un intrus de pénétrer le réseau d'une entreprise qu'Internet. Maintes sociétés ont monté des accès modem entrants (*dial-in*) pour leurs clients itinérants. Ces accès sont souvent non chiffrés et avec une authentification pauvre, et sont donc faciles à pirater, surtout depuis que l'on peut masquer le numéro de l'appelant. Jusqu'à maintenant, cela coûtait très cher de pirater ce genre de point d'entrée, car le coût des communications téléphoniques était rédhibitoire. Mais avec les récents opérateurs qui proposent le téléphone gratuit, gageons que le *war dialing* (méthode consistant à chercher des numéros de téléphone offrant un service d'accès modem entrant) va recommencer. Citons également la vieille technologie X.25 qui est encore souvent utilisée en entreprise. Les pirates russes, qui sont experts de cette technologie, réussissent souvent des intrusions au sein d'entreprises par ce biais.