Introduction au Firewall du Speed Touch Home (STH) v0.9-1

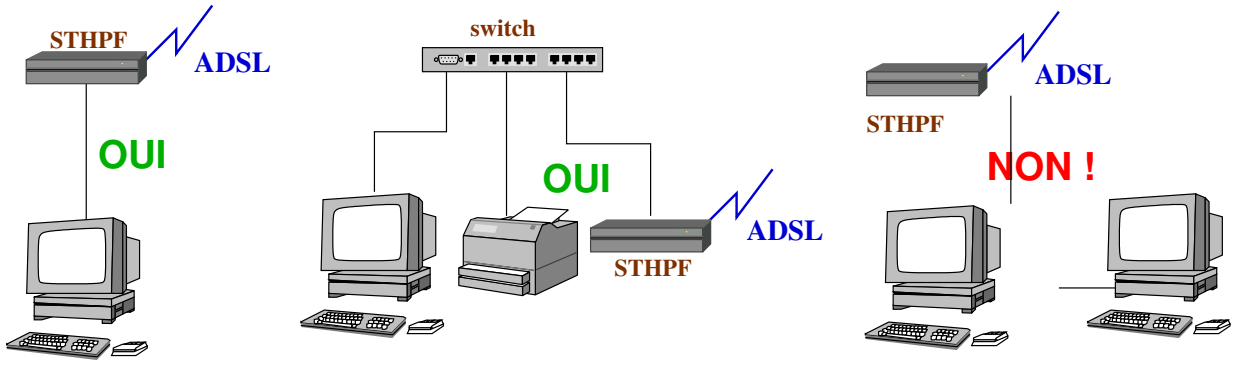
torres@ras.eu.org

De quoi que ça cause

Les détenteurs d'un « modem » ADSL Alcatel *Speed Touch Home* disposent, en vérité, d'un routeur doublé d'un firewall. Ces fonctionnalités sont tout bêtement « gelées » par programmation. Le site de MacADSL met à disposition du public, informations et conseils sur la marche à suivre pour récupérer toutes les fonctionnalités de cet équipement.

Je supposerai donc que le lecteur de cette introduction a déjà pris connaissance de la documentation disponible et qu'il a procédé au dévérouillage du routeur, puis à l'installation du firewall. Je traiterai ici, de la configuration de ce firewall. Ce n'est pas un cours sur les firewalls, ni un substitut à la documentation technique disponible sur le site de TMM (manuel utilisateur et guide de référence). Cette introduction se limite à expliquer le minimum vital à appréhender pour configurer ce firewall, sachant que la documentation technique s'adresse à des techniciens des réseaux.

Une fois dégelé, libéré, le STH devient un outil puissant, capable de s'adapter à des architectures de réseau variées (plusieurs liaisons ADSL, réseaux virtuels privés, etc.). Pour simplifier la présentation, je me limiterai à la situation d'un utilisateur grand public dont le modem ADSL est tout bonnement « relié » à son fournisseur d'accès à internet. Quant au réseau local de l'utilisateur, je supposerai uniquement qu'un ou plusieurs ordinateurs sont reliés au STH par un même réseau ou, plus exactement, que le STH et un ou plusieurs ordinateurs sont sur le même réseau.



Dans la suite, je ne parlerai plus du STH mais du STHPF : le Speed Touch Home devenu Pro Firewall. C'est ainsi que je désignerai le « modem ».

Aux utilisateurs de Windows

Vous avez installé un énorme cheval de Troie derrière le firewall, commencez par le désinstaller.

> Redémarrez votre ordinateur avec une quelconque disquette de démarrage
>
> À l'invite, tapez *format c:*
>
> Installez GNU-Linux.
>
> Vous pouvez reprendre la procédure normale.

Mon modem est un ordinateur...

Pour comprendre comment fonctionne le firewall du STHPF, il faut oublier que c'est un modem. Il vaut mieux le considérer comme un ordinateur équipé d'une carte modem ADSL et d'une carte ethernet (schéma A).

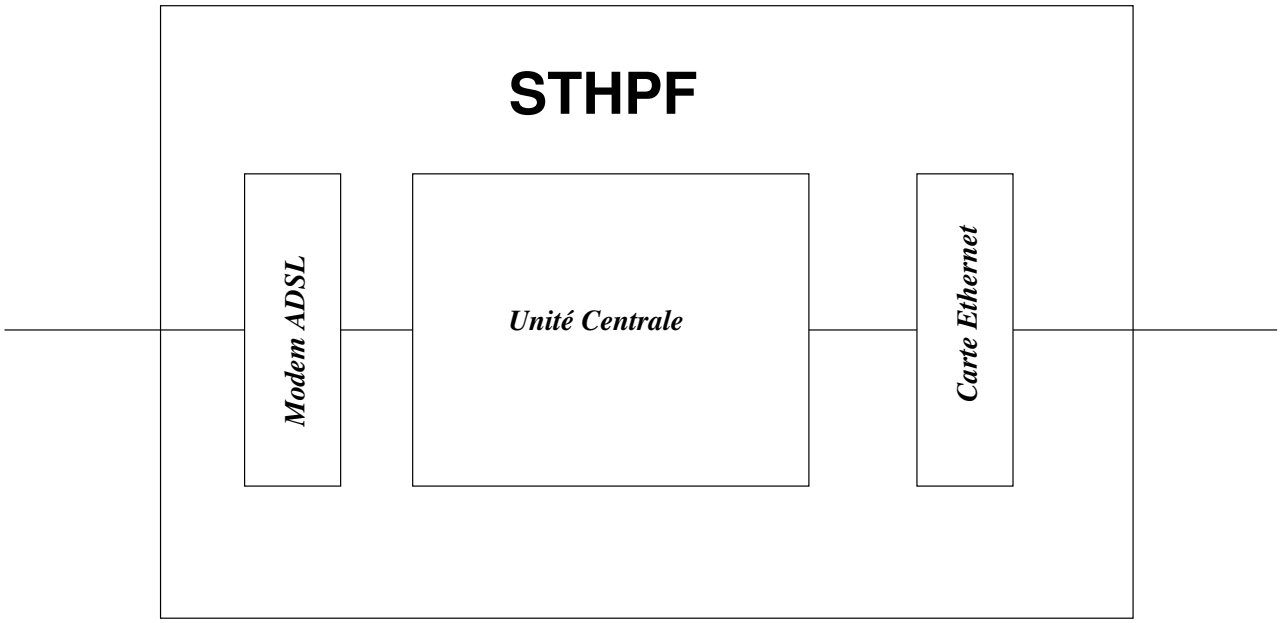


Schéma A

Le firewall ne s'occupant ni de la gestion du modem, ni de la gestion de la carte ethernet, on peut même faire abstraction de ces périphériques, pour ne s'intéresser qu'à la partie du STHPF faisant office d'ordinateur (schéma B).

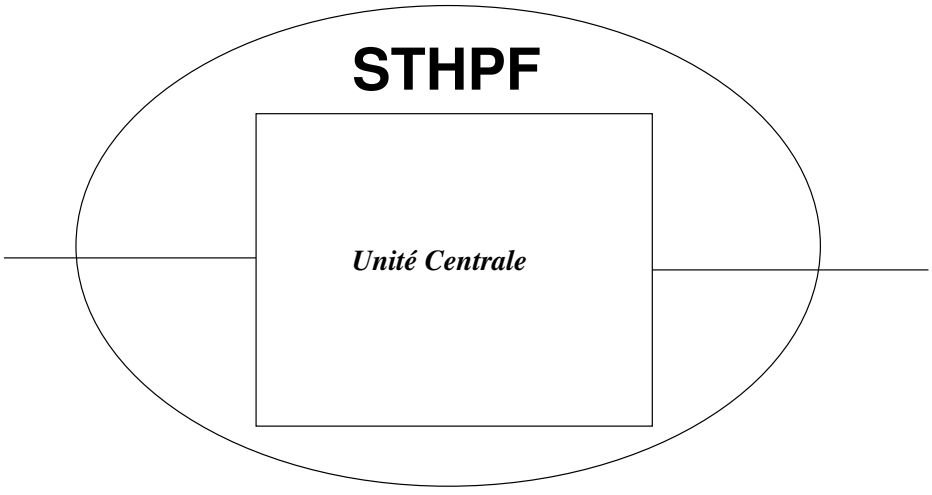


Schéma B

Au final, on considèrera le STHPF comme une passerelle entre un *réseau local* et un *réseau distant* (schéma C).

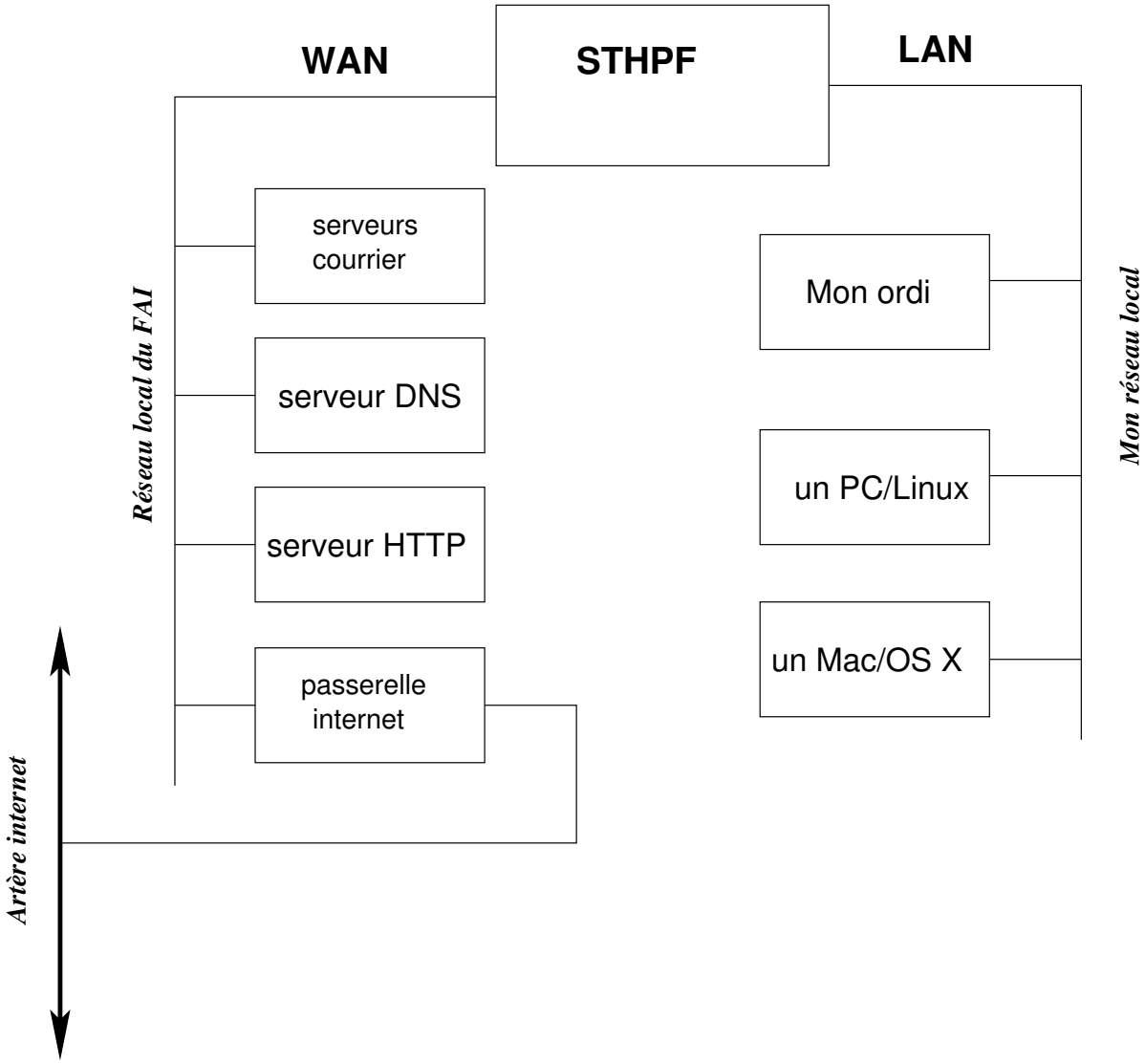


Schéma C

Le réseau local (LAN[1]) désigne les ordinateurs de l'utilisateur. Puisque l'on considère le STHPF comme un ordinateur, on peut affirmer que le plus petit réseau local contient au moins deux machines : le STHPF et l'ordi de l'utilisateur.

Le réseau distant (WAN[2]) est constitué de l'ensemble des ordinateurs accessibles à travers notre fournisseur d'accès à internet (FAI). En simplifiant, on peut considérer que le WAN est l'*internet*[3]. On remarque que le STHPF faisant partie du WAN, il fait automatiquement partie de l'internet. Ce n'est pas le cas de l'ordi de l'utilisateur.

Comme toute passerelle, le STHPF a un pied sur chaque rive qu'il relie. Il appartient aux deux réseaux et possède donc deux adresses : une adresse locale (généralement 10.0.0.138) et une adresse internet (déterminée par votre FAI). Les ordinateurs du WAN ne « voient » que son adresse internet, alors que ceux du LAN ne voient que son adresse locale.

1 LAN = Local Area Network.
2 WAN = Wide Area Network.
3 Dans notre cas, l'expression WAN, généralement utilisée pour décrire des réseaux de réseaux privés ,perd un peu de sa pertinence.

Entrer, sortir

Nous avons l'habitude de dire que les données allant du WAN vers le LAN entrent, ou descendent (schéma D1). À l'inverse, nous disons que les données allant du LAN vers le WAN sortent, on montent (schéma D2). Cela est parfaitement censé. Ce qui vient de l'extérieur « entre » et ce qui va vers l'extérieur « sort ».

Schéma D1

Schéma D2

Nous décrivons ainsi les échanges de données du point de vue particulier qui est le nôtre, celui que nous avons depuis une rive particulière. Les termes *enter* et *sortir* expriment manifestement un point de vue relatif[4].

Si l'on adopte le point de vue de la passerelle, les choses apparaissent de manière bien différente. Pour le STHPF, le LAN et le WAN sont deux réseaux totalement symétriques. Vu de l'intérieur du STHPF, les données entrantes sont celles qui proviennent de l'extérieur (schéma E1) et les données sortantes sont celles qui partent vers l'extérieur (schéma E2). L'utilisateur et le STHPF donnent bien le même sens au mots. C'est leur point de vue qui diffère. Cela oblige à une gymnastique intellectuelle, dans la mesure où, pour l'utilisateur, les deux réseaux remplissent des fonctions très distinctes.

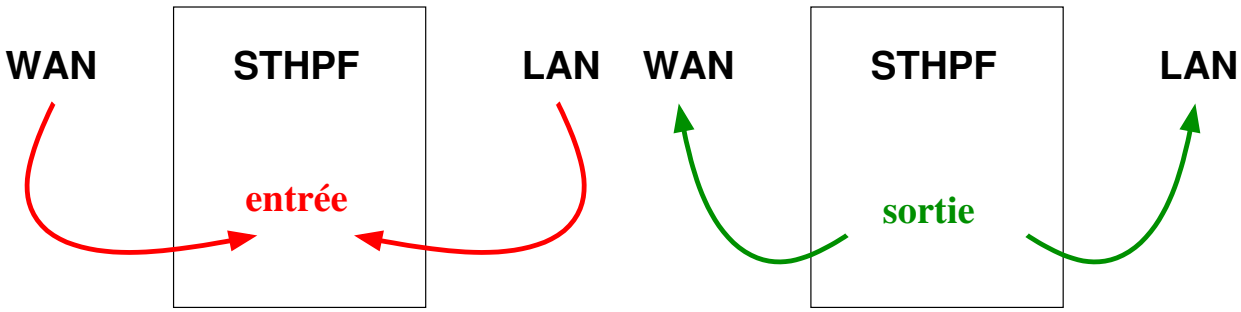


Schéma E1 Schéma E2

Ainsi, ce que l'utilisateur considère comme un flux « entrant » correspond, pour le STHPF, à deux flux simultanés : un flux entrant et un flux sortant. Ce que l'utilisateur considère comme un flux « sortant » correspond identiquement, pour le STHPF, à un flux entrant suivi d'un flux sortant. Cela ne signifie pas que le STHPF ne fait pas la différence entre le LAN et le WAN. Il sait parfaitement

4 On pense aux mots que Victor Hugo met dans la bouche de Gavroche qui, sortant d'une échoppe, s'exclame « entrons dans la rue ».

si une entrée provient du LAN ou du WAN.

Ce n'est qu'un problème de terminologie, mais il est primordial d'adopter le point de vue du STHPF si l'on veut comprendre le fonctionnement du firewall. Nous allons donc devoir reformuler un objectif de protection exprimé dans les termes de l'utilisateur (« se protéger de l'extérieur ») relativement à la position du firewall. Pour ce dernier, « l'extérieur » désigne aussi bien l'ordi de l'utilisateur que l'ordinateur d'un éventuel agresseur.

Un serveur spécialisé

Jusqu'ici, nous avons représenté le STHPF comme une boîte noire. Nous l'avons dépouillé de ses périphériques afin de nous concentrer sur l'essentiel. Le moment est venu de détailler le fonctionnement cet ordinateur spécialisé.

Étant dépourvu d'interface physique classique écran ou clavier, le STHPF n'est accessible à l'utilisateur qu'à travers des interfaces logicielles, comme n'importe quel serveur sur l'internet. Au niveau le plus général, le STHPF se présente à l'utilisateur comme la mise à disposition d'un ensemble de fonctionnalités, à travers des serveurs (schéma F) :

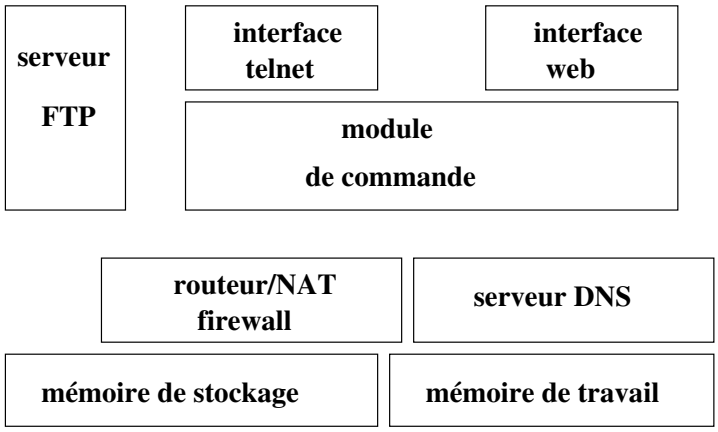


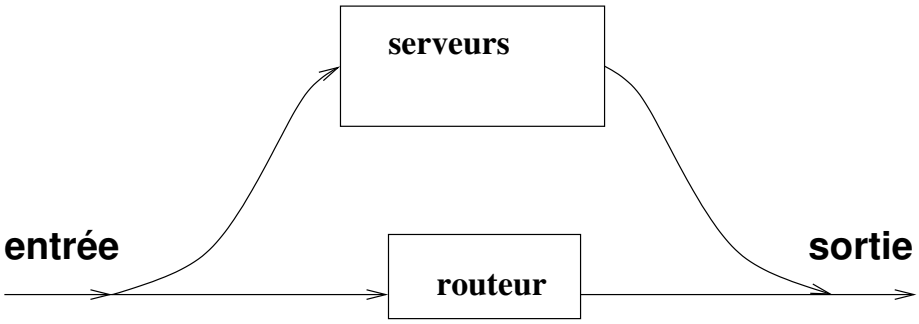
Schéma F

L'ensemble des services (FTP, Web, Telnet, DNS) sont accessibles depuis l'extérieur du STHPF, c'est-à-dire depuis le LAN ou le WAN. Sur le LAN on y accède par l'adresse locale (10.0.0.138), sur le WAN on y accède par l'adresse internet (par ex. 210.3.87.25). Bien que la majorité des utilisateurs de base n'emprunte que l'accès par le LAN, il faut garder présent à l'esprit que le STHPF est un serveur accessible par le WAN. C'est même le seul équipement du réseau local directement visible sur l'internet[5].

Le STHPF n'est donc pas une simple « boîte » de transit pour les données allant du LAN vers le WAN ou vice-versa. En tant que serveur, il est destinataire final et émetteur originel de données. On en déduit qu'une donnée entrante dans le STHPF est soit destinée à ressortir, soit destinée à être utilisée sur place (par un des serveurs). Suivant le même raisonnement, une donnée sortant du STHPF provient soit d'une donnée entrante (qui n'a fait que transiter), soit d'une donnée issue de l'un de ses serveurs internes. Cette description des flux, à l'intérieur du STHPF est résumé par le schéma G.

Schéma G

5 Puisque l'ordi de l'utilisateur en est découplé par une passerelle.



Il ne nous reste plus qu'à faire apparaître la fonction de traduction d'adresses et de ports (NAT/PAT[6]), celle-là même qui permet au STHPF de faire office de guichet public, sur l'internet, pour l'ordi de l'utilisateur. Cette fonction est utilisée dans deux situations distinctes.

Lorsqu'un ordinateur du réseau local émet une requête vers l'internet, le STHPF **endosse** cette requête, c'est-à-dire qu'il l'effectue, prétendument en son nom propre, sur l'internet. Le serveur sollicité lui adressera donc la réponse à la requête, s'imaginant que le STHPF est vraiment l'ordi demandeur. Dès la réception des données, le STHPF les transmettra à l'ordi ayant effectué la requête originelle. Cette opération de traduction au coup par coup, est appelée NAT/PAT *dynamique*.

Mais la traduction est également indispensable si l'on veut qu'un service (serveur) installé sur le LAN soit accessible depuis l'internet. Dans ce cas, le service sera identifié par l'adresse publique du STHPF et un numéro de port spécifique. Lors de la réception des requêtes venant de l'internet, le STHPF les réoriente vers l'ordi du réseau local sur lequel le service a été installé (traduction). Pour que le service soit réellement accessible, il faut que les ordinateurs demandeurs connaissent, à l'avance, le couple (adresse : port) qui identifie le service de manière unique, dans tout l'internet. Étant donné que l'adresse publique et l'adresse locale du service sont définies de manière stable, on parle de NAT/PAT *statique*. Par exemple, toutes les demandes provenant du WAN à destination du serveur web local seront adressées à 210.3.87.25 : 81 et le STHPF les transmettra à 10.0.0.1 : 80.

Nous pouvons désormais présenter un schéma faisant apparaître toutes les fonctionnalités du STHPF nécessaires à la compréhension du fonctionnement du firewall (schéma H).

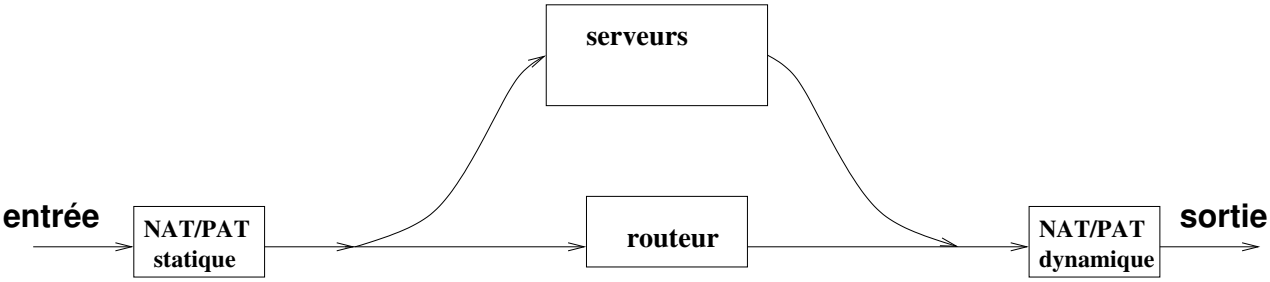


Schéma H

Le firewall du STHPF

Le terme firewall décrit un objectif et non une technique. Un firewall est un pare-feu, non une technique de pare-feu. Le firewall du STHPF s'appuie sur le principe du *filtrage* de paquets[7]. Cette technique consiste à ne rien changer au cheminement normal des paquets, tout en insérant des points de contrôle. En chaque point de contrôle, le filtrage se résume alors à bloquer ou laisser passer un paquet, sur la base d'une batterie de test satisfaits par le paquet (par exemple, l'ordi

6 NAT/PAT = Network Adress Translation/ Port Adress Translation. Autre notation équivalente : NA(P)T.

7 Le « paquet » étant la conteneur servant à transporter les données.

émetteur, le destinataire, le service, etc.). En gros, on ne modifie pas le flux de traitement mais on subtilise, au passage, les paquets non désirés.

En reprenant le schéma H, nous pouvons désormais faire figurer les 5 points de contrôle (HOOKs) permis par le firewall du STHPF (schéma I).

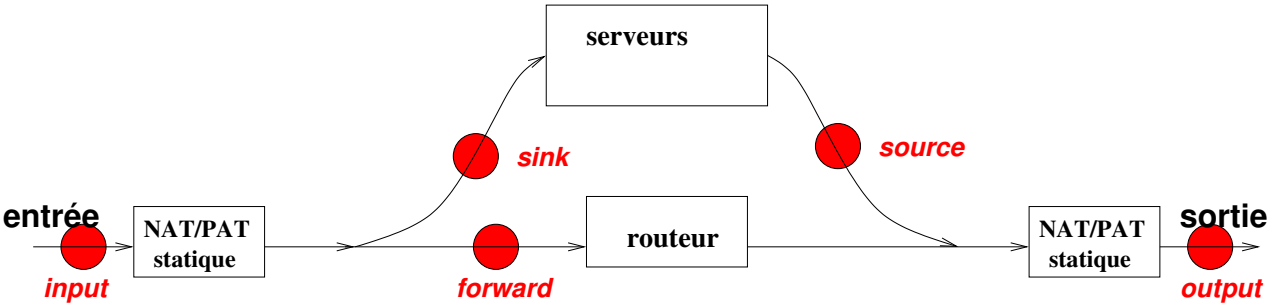


Schéma I

Le hook *input*, permet de tester tout paquet entrant dans le STHPF, préalablement à tout autre traitement. Au risque de se répéter, il n'est pas inutile de rappeler que la notion d'*entrée* étant relative au STHPF, cela recouvre tout paquet allant :

- du WAN vers le LAN
- du LAN vers le WAN
- du WAN vers les serveurs du STHPF
- du LAN vers les serveurs du STHPF.

Le hook *sink*, permet de ne considérer que les paquets destinés aux serveurs du STHPF. Comme tout paquet passant en *sink* est préalablement passé par *input*, on peut de demander à quoi sert ce hook. En effet, plutôt que de tester un paquet lorsqu'il arrive en *sink*, on aurait pu le tester dès son passage par *input*. C'est presque vrai, mais pas tout à fait... Deux raisons justifient la création du hook *sink*.

La plus importante est que les paquets arrivant en *sink* ne sont pas forcément identiques à ceux qui passent par *input*. Entre ces deux hooks, le module de NAT/PAT statique a pu modifier les paquets concernés. En particulier, certains paquets qui étaient apparemment destinés au STHPF en *input*, verront leur destinataire modifié après passage par le NAT/PAT. C'est notamment le cas de tous les paquets destinés à des serveurs du réseau local que l'on a voulu rendre accessibles sur l'internet.

Le hook *sink* se justifie également pour des raisons moins fondamentales mais non moins pratiques. Il est commode et économique de regrouper et d'isoler les test ne concernant que la protection des accès aux serveurs du STHPF. Ce point névralgique est le verrou de protection du STHPF. Rien ne sert de protéger un réseau local par un firewall, si ce firewall n'est pas lui-même protégé.

Le hook *source* permet de tester les paquets émis par les serveurs du STHPF.

Le hook *forward* permet de tester les paquets ne faisant que transiter par le STHPF (du LAN vers le LAN, du WAN vers le WAN, du LAN vers le WAN ou du WAN vers le LAN). Comme nous l'avons vu pour le hook *sink*, le hook *forward* n'est pas redondant avec le hook *input*. Les paquets lui parviennent après traduction éventuelle par le module NAT/PAT statique.

Le hook *output* permet de tester tout paquet sortant du STHPF. Comme pour le hook *input*, le terme « sortant » est à prendre dans un sens relatif au STHPF et non à l'utilisateur. Le flux sortant est donc composé :

- du flux entrant,
- diminué du flux destiné aux serveurs du STHPF,
- augmenté du flux émis par les serveurs du STHPF,
- diminué du flux éliminé par les hook précédents.

On remarquera que l'ensemble des paquets parvenant en *output* n'est pas strictement identique au regroupement des paquets franchissant les filtres *source* et *forward*. Les parquets venant de *source* ou de *forward* ont été éventuellement transformés lors de leur passage par le module NAT/PAT dynamique. De plus le passage par le module de routage apporte des informations supplémentaires[8] qui ne peuvent donc être testée qu'au hook *output*.

Paramétrer le firewall

Le paramétrage du firewall nous conduira à définir 5 batteries de tests, chacune décrivant les contrôles de paquets à effectuer au passage de l'un des 5 hooks. Le langage de commande permettant de formuler ces tests d'une manière compréhensible par le STHPF est disponible dans le manuel de référence (*CLI Reference Guide*). Nous nous limiterons ici à en présenter les principes généraux que nous illustrerons par des exemples.

a) Les contrôles sont rédigés sous forme de *règles* ayant l'organisation générale suivante :

```
si le paquet satisfait à condition1 [et condition2 [et
condition3 [et ...]]] alors (bloquer|laisser-passer) le paquet.
```

b) Les contrôles à effectuer au passage d'un hook sont décrit comme une liste numérotée de règles. Les règles sont évaluées successivement, suivant l'ordre croissant des numéros. Dans la terminologie du STHPF, la séquence de règles applicables en un hook est appelée une « chain ».

c) Dès qu'un paquet satisfait aux conditions d'une règle, on lui applique l'action indiquée par la règle (bloquer ou laisser passer); son traitement dans ce hook est terminé. Si le paquet ne satisfait pas aux conditions de la règle, on tente de lui appliquer la règle suivante. Et ainsi de suite jusqu'à ce qu'on trouve une règle qui s'applique ou qu'on ait épuisé la *chain* du hook. Dans ce dernier cas, le STHPF ne peut que prendre une décision arbitraire[9].

d) La convention retenue par les concepteurs du STHPF est que si un paquet passant par un hook ne satisfait à aucune règle, alors on le laisse passer.

Au départ, aucune règle n'étant prédéfinie, tout est autorisé. En clair, lorsqu'on active le firewall, il ne protège rien. Face à cette situation, les préceptes généraux de protection voudraient que l'on traite l'autorisation d'accès sur le mode de l'exception. Cela revient à commencer par écrire une règle[10] bloquant tous les parquets. Par la suite, on fera précéder cette règle par autant de règles que nécessaire pour traiter les « exceptions » que constituent les accès autorisés.

Exemple de règle bloquant tous les accès au STHPF... et donc à l'internet :

```
firewall rule create chain=INPUT action=drop
```

Protéger le firewall !

Inutile de mettre en place un firewall si celui-ci est vulnérable. Ce serait comme laisser la clé sur la serrure d'une porte blindée. Or, la prise de contrôle du STHPF étant accessible à travers les serveurs http, telnet et ftp internes au STHPF, elle ouverte aussi bien au WAN qu'au LAN.

Sink étant le hook situé à l'entrée des serveurs internes, c'est là que l'attention doit être portée en

8 Les informations de routage, précisément.
9 Ce qui revient à appliquer une règle par défaut, non écrite.
10 En fait, 5 règles, une par hook, que l'on placera toujours en fin de *chain*. On inverse ainsi le traitement « par défaut » de STHPF.

priorité. S'agissant de la clé de voûte du système de protection, l'accès à ces serveurs doit être limité au maximum.

Dans le cas d'un réseau local limité à deux machines (le STHPF et le poste utilisateur), la règle de protection peut s'énoncer ainsi :

on limite l'accès aux services internes du STHPF au seul poste utilisateur

Nous exprimerons la même idée en disant que tout paquet arrivant en *sink* doit être bloqué s'il vient du WAN. Rédigé en langage de commande, cela prend la forme suivante :

```
firewall rule create chain=SINK srcintf=wan action=drop
```

Attention, si vous écriviez :

```
firewall rule create chain=SINK action=drop
```

vous perdriez tout contact avec le STHPF... (si vous ne comprenez pas pourquoi, c'est que quelque chose d'important vous a échappé; relisez les parties précédentes).

Dans bien des cas, l'interdiction de tout accès provenant du WAN est trop restrictive. En particulier, si l'on utilise le serveur DNS du STHPF, cette règle le rendrait inopérant. En effet, le serveur DNS doit pourvoir recevoir des informations provenant d'autres serveurs DNS, situé dans l'internet (en particulier celui de votre FAI). Afin d'ouvrir l'accès au STHPF, uniquement aux réponses des serveurs DNS, on ajoute la règle suivante :

```
firewall rule create chain=SINK srcintf=wan prot=udp dstport=dns action=accept
```

on peut être plus précis et limiter l'accès aux réponses fournies par un unique serveur de noms de domaines :

```
firewall rule create chain=SINK srcintf=wan src=210.65.21.78/32 prot=udp dstport=dns
action=accept
```

Regroupons toutes les règles que nous voudrions voir appliquer au hook *sink*. L'attribut *index* précise l'ordre croissant dans lequel les règles doivent être examinées. Au final, voilà à quoi pourrait ressembler la chain du hook *sink* :

```
firewall rule create chain=SINK index=0 srcintf=eth0 srcbridgeport=1 action=accept
firewall rule create chain=SINK index=2 prot=udp dstport=dns action=accept
firewall rule create chain=SINK index=5 action=drop
```

Pour comprendre la signification des ces trois règles, nous allons les examiner dans l'ordre inverse de leur ordre d'évaluer par le STHPF. Nous faisons cela, car le seul moyen de connaître le rôle joué par une règle « accept » est de savoir comment seront traités les paquets qu'elle laisse passer.

La règle 5 (index=5) inverse le traitement par défaut des paquets, tel que programmé dans le STHPF. Elle indique que tout paquet qui n'a pas été explicitement « reconnu » (par l'une des règles précédentes) doit être rejeté. C'est l'application du principe d'*autorisation par exception*.

La règle 2 autorise tout paquet destiné au dns. Cela concerne aussi bien les requêtes du réseau local que les réponses des serveurs dns de l'internet. En réalité, seuls les paquets venant de l'internet seront confrontés à cette règle car les paquets dns venant du réseau local sont déjà interceptés par la règle 0 (voir plus loin). On voit ici, que la signification d'une règle dépend de son contexte. Autrement dit, en ajoutant (ou en ôtant) une règle à une *chain*, on change non seulement le sens global de la *chain*, mais aussi, potentiellement, la signification individuelle des chacune des autres règles de la *chain*.

La règle 0 autorise tout paquet provenant du réseau local. En fait, elle autorise, depuis l'ordi de l'utilisateur, tout type d'accès aux serveurs web, ftp, telnet, dns.

Énonçons nos trois règles en jargon informatique courant :

- règle 0 : l'utilisateur a tous les droits
- règle 2 : seul l'accès dns est autorisé depuis l'internet
- règle 5 : tout ce qui n'est pas autorisé par l'une des règles précédentes est interdit.

Dans cet exemple, les numéros de règles ne se suivent pas.. C'est parfaitement « légal » puisque la seule chose qui importe est l'ordre entre les règles. Lors de la saisie, si aucun numéro n'est précisé, la règle prend automatiquement l'index 0. S'il existe déjà une règle d'index 0, l'ancienne règle 0 verra son index augmenté de 1 ; et ainsi de suite, s'il exite déjà une règle 1,2,3 ... Saisir les règles sans numéro revient donc à les empiler les unes sur autres. La première règle saisie, sera la dernière règle évaluée puisque, au final, est se retrouvera en queue de chain. On retrouve donc, à la saisie, la logique (inversée) signalée au début de commentaire.

Protection du réseau local

Une fois le firewall sécurisé on peut revenir à notre objectif premier : protéger le réseau local. Le problème n'est pas tant de paramétrer le firewall que de trouver des restrictions pertinentes à appliquer... En effet, quelle protection supplémentaire va-t-on demander au firewall ? N'oublions pas que le routage NAT/PAT a déjà découplé le réseau local de l'internet...

Il y a donc une précaution à prendre, c'est précisément d'éviter qu'un ordinateur de l'internet ne se fasse passer pour un ordinateur du réseau local (IP spoofing[11]). Sinon, ce que l'on a autorisé aux postes locaux (c'est-à-dire *tout*, cf. règle 0) serait accessible à n'importe qui. On déjouera cette forme d'*IP spoofing* en appliquant un contrôle dès l'entrée dans le STHPF, au hook *input*.

```
firewall rule create chain=INPUT srcintf=wan src=10.0.0.0/8 action=drop
```

Pour le reste, ça n'a pas grand sens de limiter notre propre accès à certains services. Nous ne sommes pas dans la position d'un administrateur de réseau qui définit les privilèges de chaque poste de travail en fonction de profils d'usages. Nous n'avons qu'un poste (le-nôtre-à-nous-perso) et nous en sommes le super-utilisateur[12]. Il est donc logique que nous ne souffrions aucune restriction, tant dans nos accès à l'internet qu'au STHPF.

Les seules limitations intéressant un utilisateur individuel concerneront éventuellement les restrictions d'accès à ses propres serveurs. L'utilisation d'un firewall sur la passerelle permet d'interdire l'accès aux indésirables, avant leur entrée sur le réseau local. Il est ainsi beaucoup plus fiable de bloquer l'accès à un serveur http dans la passerelle qu'au niveau d'un poste utilisateur. Tout n'est pas résolu pour autant, puisque le STHPF est incapable de déterminer si l'adresse d'expédition inscrite sur le paquet est réellement l'adresse de l'expéditeur ou de son mandataire[13].

Malgré nous

On pourra encore appliquer quelques règles de bonne politique visant à empêcher que notre réseau local ou notre routeur ne puisse servir de relais à des attaques vers d'autres machines.

Ainsi, on refusera de réexpédier des paquets provenant de l'internet et destinés à l'internet[14] :

```
firewall rule create chain=OUTPUT index=0 srcintfgrp=wan dstintfgrp=wan action=drop
```

11 Tromperie sur l'adresse IP.
12 Comprenez : le « Dieu » d'un tas de sable, cent fois plus crétin qu'une mouche.
13 Seuls les protocoles sécurisés le permettent mais il est pratiquement inenvisageable de s'autolimiter à ceux-là.
14 En effet, si vous regardez le paramétrage de votre routeur (par l'interface Web) vous constaterez que votre STHPF injecte sur l'internet tout ce qu'il reçoit, dès lors que ça n'est destiné au réseau local ou à lui-même.

Les statistiques du firewall

En remontant la protection sur la passerelle, le firewall du STHPF offre plus de fiabilité qu'un logiciel de filtrage de paquet, implanté sur le poste utilisateur (même unique). En revanche, il prive l'utilisateur des journaux, synthèses et statistiques fournies par ce type de logiciel. Or, l'observation de l'agresseur est une composante essentielle de toute doctrine de protection. Il est donc plus que conseillé d'utiliser les maigres outils d'information proposés par le STHPF.

Le firewall du STHPF tient à jour une comptabilité des règles déclenchées. Il est toujours instructif de les consulter. Les règles « drop » vous indiquerons le nombre impressionnant de paquets non désirés charriés aux portes de votre réseau local, par un accès permanent à l'internet. Vous constaterez notamment que l'IP spoofing se porte bien...

Mais la surveillance des statistiques n'a pas pour but de nous renforcer dans nos certitudes... au contraire. C'est pourtant la seule chose que l'on peut faire à partir de règles « drop ». Quant aux statistiques des règles « accept », elles nous indiquent à quel point ce qui se passe bien, va bien... Heureusement, il existe un troisième type de règle (*count*), dont leur seul but est de compter les déclenchement de règles. Lorsqu'un paquet vérifie les conditions d'une règle *count*, l'évènement est comptabilisé mais le paquet continue à être confronté aux règles suivante de la *chain*.

Voici un exemple :

```
firewall rule create chain=SINK index=4 srcintfgrp=wan prot=tcp dstport=4662 action=count
firewall rule create chain=SINK index=5 action=drop
```

Ici, même les paquets vérifiant la règle 4 seront transmis à la règle 5, pour traitement. Le seul but de la règle 4 est donc de recueillir des informations sur la nature des paquets que la règle 5 rejette, en bloc.

En supposant que vous n'ayez pas eu l'idée saugrenue d'ouvrir un serveur eDonkey sur votre LAN, elle vous indiquera le nombre de tentatives d'accès (attaques ?) à un hypothétique serveur. Il ne faut pas oublier qu'à l'exception des ports ouvert en NAT/PAT statiques, toutes les requêtes faite sur votre adresse internet (depuis tout l'internet) aboutissent naturellement en *sink*.

Liens intéressants

Les codes numériques des services

Comme on l'a vu dans les exemples, il peut s'avérer indispensable de filtrer les paquets en fonction du service concerné. Celui-ci est généralement identifié par un doublet protocole:port. Le STHPF connaît les ports assigné aux protocoles les plus courants, mais pas tous. Ainsi, on a pu signifier le *dstport* normalisé du DNS en entrant la chaîne de caractères « dns ». Qu'en est-il des autres protocoles ? Il existe une liste officielle donnant les valeurs numériques de tous les protocoles officiellement enregistrés.

Bien sûr, si vous avez « cassé » votre accès internet en jouant avec votre firewall, il est trop tard pour vous connecter au site officiel. Les utilisateurs de GNU-Linux pourront toujours se reporter au fichier texte /etc/services.