IP Masquerade

principe et mise en place

EBC Informatique

Centre de compétences UNIX

TABLE DES MATIERES

Notes préliminaires :

Tout en tentant de rester le plus général possible dans cette étude, je tiens à vous informer que la totalité des manipulations UNIX expliquées ont été effectuées sur un serveur fonctionnant sous distribution RedHAT 7.0 disposant du noyau 2.2.16.

Avec cette distribution, IP Chains est à sa version 1.3.9. Les versions plus récentes ne devrait pas poser de problèmes particulier par rapport aux opérations de cette étude. Les sources, ainsi que les binaires du programmes, sont disponibles sur la plupart des sites de téléchargements de programmes GNU tel : http://www.freshmeat.net/.

Concernant les systèmes Windows, j'ai testé l'ensemble des configurations réseau décrites avec Microsoft Windows 98 et Microsoft Windows NT 4 (SP 6).

I] Introduction : qu'est ce que l'IP Masquerade ?

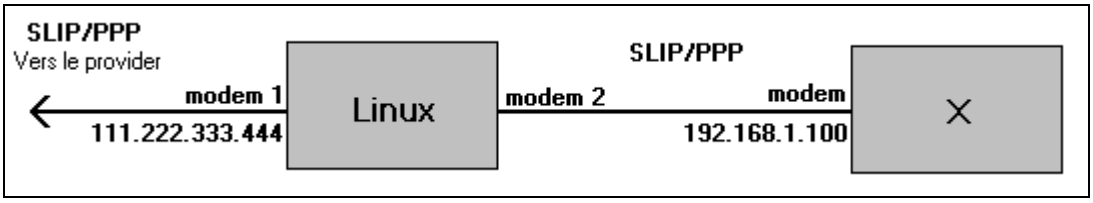
L'IP Masquerade est une fonctionnalité réseau de Linux. Si un hôte Linux est connecté à Internet avec l'option IP Masquerade en place, alors les ordinateurs se connectant à celui-ci (que cela soit sur le réseau local ou par modem) peuvent également atteindre Interne, même s'il n'ont pas d'adresse IP officielle.

Cela permet à un ensemble de machines d'accéder de manière invisible à Internet, caché derrière une passerelle, qui apparaît comme étant le seul système utilisant la connexion Internet. Il devrait être énormément plus difficile de contourner un système basé sur le masquerade, s'il est bien configuré, que de passer outre un bon firewall effectuant du filtrage de paquets (en supposant qu'il n'y a de bogues chez aucun des deux).

II] Comment fonctionne l'IP Masquerade ?

(D'après la FAQ IP Masquerade, de Ken Eves)

Voici un schéma du plus simple cas possible :



Dans le schéma ci-dessus, un ordinateur sous Linux, utilisant ip_masquerading est connecté à Internet par un lien SLIP ou PPP, utilisant le modem 1. Il possède l'adresse IP (officielle) 111.222.333.444. Il est configuré de telle façon que le modem 2 permet aux appelants de se connecter et d'initier une connexion PPP ou SLIP.
Le second système (qui n'utilise par forcément Linux comme système d'exploitation) se connecte par modem sur l'hôte Linux et entame une liaison SLIP ou PPP. Il ne possède pas d'adresse IP officielle, dans notre exemple, il utilise 192.168.1.100.

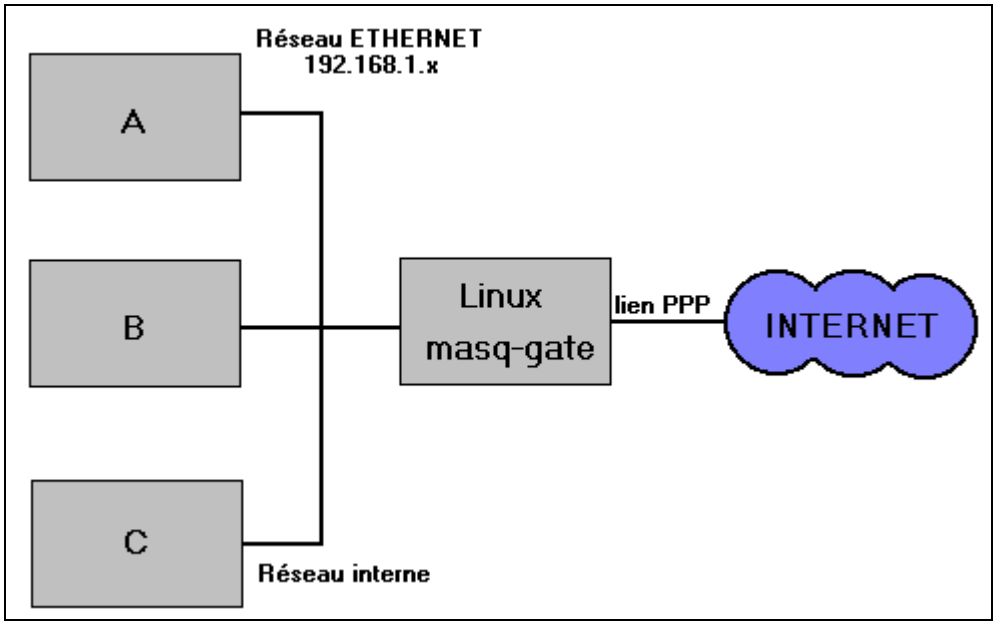
Avec l'option ip_masquerade et un routage configuré correctement, la machine X peut interagir avec Internet comme si elle était réellement connectée (à quelques exceptions près).

Pour citer Pauline Middlelink : N'oublie pas de rappeler que la machine X doit déclarer l'hôte Linux comme passerelle (que cela soit la route par défaut ou juste un sous réseau importe peu). Si X ne peut pas le faire, l'hôte Linux devra faire du proxy arp pour toutes les adresses routées, mais la mise en place du proxy arp est hors du domaine de ce document…

Ce qui suit est l'extrait d'un article de news Internet qui a été modifié pour utiliser les noms des machines de l'exemple ci-dessus :
- J'indique à la machine X que le serveur Linux est sa passerelle.
- Quand un paquet en provenance de X arrive sur la machine Linux, elle va lui assigner un nouveau numéro de port, et indiquer sa propre adresse IP dans l'entête du paquet, tout en sauvegardant l'entête originale. Elle va alors envoyer le paquet modifié à travers son interface SLIP ou PPP, vers Internet.
- Lorsqu'un paquet en provenance d'Internet arrive sur la machine Linux, si le numéro de port est un de ceux assignés à l'étape précédente, elle va modifier à nouveau l'entête pour y remettre les numéros de port et adresses IP originaux, et alors envoyer le paquet à la machine X.
- L'hôte qui a envoyé le paquet ne verra jamais la différence.

Un exemple d'IP Masquerading :

Voici ci-dessous le schéma d'un exemple classique :



Il y a dans cet exemple 4 ordinateurs qui nous intéressent (il y a sûrement sur la droite quelque chose sur laquelle aboutit notre connexion, et encore plus à droite une autre machine avec laquelle nous échangeons des données). L'ordinateur Linux masq-gate est la passerelle qui effectue le masquerading pour le réseau interne des ordinateurs A, B, et C, afin de les relier à Internet. Le réseau interne utilise une des adresses assignées des réseaux privés, à savoir dans ce cas le réseau de classe C 192.168.1.0, l'ordinateur Linux ayant l'adresse 192.168.1.1 et les autres ordinateurs ayant d'autres adresses sur ce réseau.

Les trois machines A, B et C (qui peuvent utiliser n'importe quel système d'exploitation, du moment qu'elles utilisent IP - comme par exemple Windows 95, Macintosh MacTCP ou même un autre Linux) peuvent se connecter à n'importe qu'elle machine sur Internet, mais masq-gate convertit toutes leurs connexions de façon à ce qu'elles semblent provenir de masq-gate, et s'arrange pour que toutes les données revenant d'Internet retournent au système qui en est à l'origine. Ainsi, les ordinateurs du réseau interne voient une route directe vers Internet et ne sont pas au courant du fait que leurs données ont été "masqueradées".

III] Mise en place d'IP Masquerade

Si votre réseau privé contient des informations vitales, repensez y à deux fois avant d'utiliser IP Masquerade. Cela constitue une passerelle pour vous, pour atteindre Internet, mais la réciproque est vraie et quelqu'un sur Internet pourrait pénétrer sur votre réseau privé.

1) Compiler le noyau pour le support d'IP Masquerade

Si votre distribution de Linux possède déjà les fonctionnalités nécessaires et les modules de compilés (la majorité des noyaux modulaires actuel auront ce dont vous avez besoin) mentionnés ci-dessous, alors vous n'avez pas à recompiler le noyau. La lecture de cette section est quand même largement recommandée car elle contient aussi d'autres informations très utiles.

a) Compiler un noyau Linux 2.2.x

Tout d'abord, Vous avez besoin des sources du noyau 2.2.x disponibles sur http://www.kernel.org/. Si c'est la première fois que vous compilez un noyau, ne vous inquiétez pas, en fait, c'est assez facile et tout est expliqué clairement dans le Linux Kernel HOWTO disponible avec les sources.

Décompressez les sources du noyau dans /usr/src/ avec la commande :

```
tar xvzf linux-2.2.x.tar.gz -C /usr/src
```

où x est la version du noyau dont vous disposez.

Assurez vous qu'il existe un répertoire ou un lien symbolique appelé linux pointant sur ce nouveau répertoire.
Appliquez les patchs appropriés. Comme de nouveaux patchs sortent régulièrement, les détails ne sont pas décrits ici. Référez vous à IP Masquerade Ressources pour des informations au jour le jour. Référez vous au Kernel HOWTO et au fichier README des sources du noyau pour plus d'informations sur la compilation d'un noyau.

Voici les options que vous devez compiler :
Répondez YES aux options suivantes :

* Prompt for development and/or incomplete code/drivers
    CONFIG_EXPERIMENTAL
- Ceci vous permettra de selectionner le code experimental IP Masquerade.

* Enable loadable module support
    CONFIG_MODULES
- Vous permet de charger les modules ipmasq tel ip_masq_ftp.o

* Networking support
    CONFIG_NET

* Network firewalls
    CONFIG_FIREWALL

* TCP/IP networking
    CONFIG_INET

* IP: forwarding/gatewaying
CONFIG_IP_FORWARD

* IP: firewalling
CONFIG_IP_FIREWALL

* IP: masquerading
CONFIG_IP_MASQUERADE

* IP: ipportfw masq support
CONFIG_IP_MASQUERADE_IPPORTFW
- Recommandé

* IP: ipautofw masquerade support
CONFIG_IP_MASQUERADE_IPAUTOFW
- Optionnel

* IP: ICMP masquerading
CONFIG_IP_MASQUERADE_ICMP
- Support pour masquerader les paquets ICMP, recommandé

* IP: always defragment
CONFIG_IP_ALWAYS_DEFRAG
- Chaudement recommandé

* Dummy net driver support
CONFIG_DUMMY
- Recommandé

* IP: ip fwmark masq-forwarding support
CONFIG_IP_MASQUERADE_MFW
- Optionnel

NOTE : Voici juste les composants qu'il faut pour que l'IP Masquerade fonctionne, sélectionnez ensuite les autres options spécifiques dont vous avez besoin pour votre système.

Après avoir compilé le noyau, vous devriez compiler et installer les modules :

```
make modules; make modules_install
```

b) Compiler un noyau 2.0.x

Vous devez tout d'abord disposer des sources du noyau (de préférence la derniere version 2.0.36 ou plus récente). Si c'est la première fois que vous compilez votre noyau, ne soyez pas effrayé. En fait, c'est plûtot simple, et tout est expliqué dans le Linux Kernel HOWTO.

Décompressez les sources du noyau dans /usr/src/ avec la commande :

```
tar xvzf linux-2.0.x.tar.gz -C /usr/src
```

où x est la version du noyau dont vous disposez.

Assurez vous qu'il existe un répertoire ou un lien symbolique appelé linux pointant sur ce nouveau répertoire.
Appliquez les patchs appropriés. Comme de nouveaux patchs sortent régulièrement, les détails ne sont pas décrits ici. Référez vous à IP Masquerade Ressources pour des informations au jour le jour. Référez vous au Kernel HOWTO et au fichier README des sources du noyau pour plus d'informations sur la compilation d'un noyau.

Voici les options que vous devez compiler :
Dites YES aux options suivantes :

* Prompt for development and/or incomplete code/drivers
    CONFIG_EXPERIMENTAL
- Ceci vous permettra de selectionner le code experimental IP Masquerade.

* Enable loadable module support
    CONFIG_MODULES
- Vous permet le chargement des modules

* Networking support
    CONFIG_NET

* Network firewalls
    CONFIG_FIREWALL

* TCP/IP networking
    CONFIG_INET

* IP: forwarding/gatewaying
    CONFIG_IP_FORWARD

* IP: firewalling
    CONFIG_IP_FIREWALL

* IP: masquerading (experimental)
    CONFIG_IP_MASQUERADE
- Bien que cela soit expérimental, il *FAUT* l'intégrer

* IP: ipautofw masquerade support (expérimental)
    CONFIG_IP_MASQUERADE_IPAUTOFW
- Recommendé

* IP: ICMP masquerading
    CONFIG_IP_MASQUERADE_ICMP
- Support pour masquerader les paquets ICMP, optionnel

* IP: always defragment
    CONFIG_IP_ALWAYS_DEFRAG
- Chaudement recommandé

* Dummy net driver support
    CONFIG_DUMMY
- Recommandé

NB : Ce sont juste les composants dont vous avez besoin pour l'IP Masquerade. Ajoutez ensuite toutes autres options nécessaire pour votre configuration personnelle.

Une fois le noyau compilé, compilez et installez les modules :

```
make modules; make modules_install
```

c) Vérifier les modules installés sur votre système

Pour obtenir la liste des modules installés sur votre système, tapez la commande suivante :

```
[root@redhat src]# modprobe -l
/lib/modules/2.2.16-22enterprise/fs/autofs.o
/lib/modules/2.2.16-22enterprise/fs/binfmt_aout.o
/lib/modules/2.2.16-22enterprise/fs/binfmt_java.o
/lib/modules/2.2.16-22enterprise/fs/binfmt_misc.o
…
/lib/modules/2.2.16-22enterprise/usb/usbkbd.o
/lib/modules/2.2.16-22enterprise/usb/usbmouse.o
/lib/modules/2.2.16-22enterprise/usb/wacom.o
/lib/modules/2.2.16-22enterprise/usb/wmforce.o
```

Pour obtenir la liste des modules relatifs à l'IP Masquerade, tapez la commande suivante :

```
[root@redhat src]# modprobe -l | grep ip_masq
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_autofw.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_cuseeme.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_ftp.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_irc.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_mfw.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_portfw.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_pptp.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_quake.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_raudio.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_user.o
/lib/modules/2.2.16-22enterprise/ipv4/ip_masq_vdolive.o
```

2) Activer l'IP Masquerade

a) Le script d'activation

Voici un script permettant d'initialiser correctement l'IP Masquerade.

```
#!/bin/sh
#
# rc.firewall – script d'initialisation de l'IP Masquerade
#
# Chargement des modules IP_MASQ requis
#
#   NOTE:  Chargez seulement les modules nécéssaires, la liste ci-dessous
#          indique la plupart d'entre eux, mais ne les charger pas.

# Initialisation de l'ensemble des modules
/sbin/depmod –a

# Support du masquerading des transferts de fichiers FTP
/sbin/modprobe ip_masq_ftp

# Support du masquerading de RealAudio sur UDP
```

```
#/sbin/modprobe ip_masq_raudio

# Support du masquerading des transferts de fichiers IRC DCC
#/sbin/modprobe ip_masq_irc

# Support du masquerading pour les réseaux Quake et QuakeWorld
#   NOTE:  Si vous rencontrez des problèmes en chargeant le module Quake, vous
#          utilisez un ancien Kernel Linux, faites le évoluer.
# (Quake I / QuakeWorld (ports 26000 and 27000))
#/sbin/modprobe ip_masq_quake

# Pour Quake I/II/III/QuakeWorld (ports 26000, 27000, 27910, 27960)
#/sbin/modprobe ip_masq_quake 26000,27000,27910,27960

# Support du masquerading du « CuSeeme video conferencing software »
#/sbin/modprobe ip_masq_cuseeme

# Supports du masquerading du « VDO-live video conferencing software »
#/sbin/modprobe ip_masq_vdolive

#ATTENTION:  Activation de l'IP Masquerade (désactivé par défaut)
#
#            Utilisateur RedHAT: vous pouvez changer cette option dans le
#                                fichier /etc/sysconfig/network en changeant:
#
#                                   FORWARD_IPV4=false
#                                        par
#                                   FORWARD_IPV4=true
echo "1" > /proc/sys/net/ipv4/ip_forward

#ATTENTION:  Active automatiquement la fragmentation IP (désactivé par défaut)
echo "1" > /proc/sys/net/ipv4/ip_always_defrag

# Utilisateur avec IP dynamique:
#
#   Si vous obtenez dynamiquement votre adresse IP par SLIP, PPP, ou DHCP, vous
#   pouvez activé cette option. Elle vous simplifiera la vie avec des
#   programmes tel que Diald.
#echo "1" > /proc/sys/net/ipv4/ip_dynaddr

# Active le patch LooseUDP dont certains jeux on-line nécéssite
#echo "1" > /proc/sys/net/ipv4/ip_masq_udp_dloose

# MASQ timeouts
#
# 2 heures de timeout pour les timeouts de sessions TCP
# 10 secondes de timeout pour le traffic après réception du paquet TCP/IP "FIN"
# 160 secondes de timeout pour le traffic UDP (important pour l'IRC)
/sbin/ipchains -M -S 7200 10 160

# DHCP:  pour les systèmes qui obtiennent leur adresse IP d'un serveur DHCP ou
#        BOOTP comme pour l'ADSL ou le cable, il est nécéssaire d'utiliser
#        l'option suivante avant la commande DENY.
#        Le nom " bootp_netif" devrait être remplacé par le nom de
#        l'interface sur laquelle le serveur DHCP ou BOOTP va donner une
#        adresse. Ce devrait être quelque chose comme "eth0", "eth1", ...
#/sbin/ipchains -A input -j ACCEPT -i bootp_netif -s 0/0 67 -d 0/0 68 -p udp

# Configuration des règles d'IP Forwarding

# ==> INSEREZ ICI VOS REGLES DE FORWARDING (VOIR X) <==
```

b) Gestion de ce script d'activation

Personnellement, sur un système Red HAT 7.0, j'ai recopié ce script dans le répertoire /etc/rc.d/ et je l'ai nommé en « rc.firewall ».

Il s'agit ensuite de rendre ce script exécutable uniquement par root en tapant la série de commande suivante :

```
[root@redhat rc.d]# chmod 744 rc.firewall
[root@redhat rc.d]# chgrp root rc.firewall
[root@redhat rc.d]# chown root rc.firewall
```

Il s'agit ensuite d'exécuter ce script à chaque démarrage. Si vous utilisez une Red HAT ou tout autre système Linux similaire (Mandrake, …), il sufit de rajouter le chemin complet d'accés au fichier à la fin du fichier /etc/rc.d/rc.local afin que le système exécute le fichier au démarrage. Dans le cas d'une Suze, c'est à la fin du fichier boot.local qu'il vous faudra rajouter cette ligne.

3) Gestion des clients du réseau

a) Assignation d'adresse IP pour le réseau local

Puisque toutes les clients n'ont pas d'adresses IP officielles, il faut leur en allouer de manière intelligente.

Selon la FAQ d'IP Masquerade :

Il existe un RFC (#1597) qui indique quelles adresses IP assigner à un réseau non connecté. Il existe 3 plages réservées spécialement à cet effet. Une de celles que j'utilise est un sous réseau de classe C, faisant partie de la plage allant de 192.168.1.n à 192.168.255.n.

Selon le RFC 1597 :

Section 3 : Adressage de réseaux privés

L' "Internet Assigned Numbers Authority" (IANA) a réservé les 3 plages suivantes pour leur utilisation par des réseaux privés :

Début de la plage	Fin de la plage
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

Nous ferons référence à la première en tant que la "plage de 24 bits", la deuxième comme "plage de 20 bits" et la troisième comme "plage de 16 bits".

Notez que la première plage n'est rien d'autre qu'un réseau de classe A, la deuxième un ensemble de 16 réseaux de classe B contigus, et la troisième un ensemble de 255 réseaux de classe C contigus.

Ainsi, si vous utilisez un réseau de classe C, vous devrez utiliser les adresses IP 192.168.n.1, 192.168.n.2, 1.92.168.n.3,… , 192.168.n.x. Dans la suite des exemples, j'utiliserais le réseau de classe C d'adresse IP 192.168.1.x.

192.168.1.1 est habituellement la machine passerelle, qui est ici votre machine Linux se connectant à Internet. Remarquez que 192.168.1.0 et 192.168.1.255 sont respectivement les adresses de réseau et de broadcast, qui sont réservées. Evitez d'utiliser ces adresses sur vos machines.

b) Configurer les client du réseau

En plus d'affecter les adresses IP pour chaque machine, vous devrez indiquer la bonne passerelle. En général, c'est plutôt simple. Vous entrez juste l'adresse de votre machine Linux (généralement 192.168.1.1) en tant qu'adresse de passerelle.

Pour le DNS, vous pouvez utiliser n'importe quel DNS utilisable. Le plus simple est d'utiliser celui qu'utilise votre machine Linux. Vous pouvez aussi, si vous le désirez, ajouter des suffixes d'ordre de recherche DNS.

Une fois configurées ces adresses IP, n'oubliez pas de relancer les programmes concernés, ou de rebooter vos machines.

Les instructions de configuration qui suivent supposent que vous utilisez un réseau de classe C, et que votre machine Linux a pour adresse 192.168.1.1. Notez que 192.168.1.0 et 192.168.1.255 sont réservées.

Configurer Windows 9x :

Si vous n'avez pas installé votre carte réseau et son driver, faites le maintenant.
Allez dans Panneau de configuration / Réseau.
Ajoutez le protocole TCP/TP si ce n'est pas déjà fait.
Dans les propriétés de TCP/IP, allez dans Adresse IP et entrez votre adresse IP, 192.168.1.x (1<x<255). Fixez le masque de sous réseau à 255.255.255.0.
Ajoutez 192.168.1.1 dans Passerelle.
Dans Configuration / Ordre de recherche DNS, ajoutez le DNS qu'utilise votre machine Linux (que l'on peut trouver dans /etc/resolv.conf). Vous pouvez éventuellement ajouter les suffixes de domaine adéquats.
Laissez les autres paramètres tels quels, à moins que vous sachiez ce que vous faites.
Cliquez sur OK dans toutes les boîtes de dialogue et relancez le système.
« Pingez » votre machine Linux pour tester la connexion réseau : Démarrer / Exécuter, tapez :

```
ping 192.168.1.1
```

(C'est seulement un test de connexion locale, vous ne pouvez pas encore pinger l'extérieur).

Vous pouvez éventuellement créer un fichier HOSTS dans le répertoire de Windows, pour que vous puissiez utiliser les noms d'hôtes des autres machines de votre réseau local. Il y a un exemple nommé « HOSTS.SAM » dans le répertoire Windows.

Configurer Windows pour Workgroup 3.11 :

Si vous n'avez pas encore installé votre carte réseau et son driver, faites le maintenant.
Installez le package TCP/IP 32b si ce n'est pas déjà fait.
Dans Groupe Principal / Installation / Configuration réseau, cliquez sur Drivers.
Séléctionnez Microsoft TCP/IP-32 3.11b dans la section Drivers Réseaux. Choisissez Configuration.

Saisissez l'adresse IP 192.168.1.x (1<x<255), et positionnez le masque de sous réseau à 255.255.255.0 et la passerelle par défaut à 192.168.1.1.
Ne sélectionnez pas Configuration automatique DHCP et mettez n'importe quoi dans la case Server WINS, à moins que vous ne fassiez partie d'un domaine Windows NT et que vous sachiez ce que vous faites.

Cliquez sur DNS, et remplissez les informations appropriés, mentionnées plus haut dans cette section. Cliquez sur OK une fois que c'est fini.

Cliquez sur Configuration avancée, cochez Utiliser le DNS pour la résolution de noms, et Utiliser LMHOSTS si vous utilisez un fichier de résolution, comme celui mentionné ci-dessus.

Cliquez alors sur OK sur toutes les boites de dialogue, et redémarrez le système.

« Pingez » votre machine Linux pour tester la connexion réseau : Démarrer / Executer, tapez :

```
ping 192.168.1.1
```

(C'est seulement un test de connexion locale, vous ne pouvez pas encore pinger l'extérieur).

Configurer Windows NT :

Si vous n'avez pas encore installé votre carte réseau et son driver, faites le maintenant.
Allez dans Groupe Principal / Panneau de configuration / Réseau.
Ajoutez le protocole TCP/IP et les composants qui s'y rattachent depuis le menu Ajout de logiciels si vous n'avez pas encore installé le service TCP/IP.
Dans la section Logiciel et carte réseau, sélectionnez Protocole TCP/IP dans la boite de choix Logiciels réseaux installés.
Dans Configuration TCP/IP, sélectionnez l'adaptateur réseau appropriées, par exemple [1]Novell NE2000 Adapter. Entrez l'adresse IP 192.168.1.x (1<x<255), positionnez le masque de sous réseau sur 255.255.255.0 et la passerelle par défaut à 192.168.1.1.
Ne sélectionnez pas Configuration automatique DHCP et mettez n'importe quoi dans la case Server WINS, à moins que vous ne fassiez partie d'un domaine Windows NT et que vous sachiez ce que vous faites.
Cliquez sur DNS, et remplissez les informations appropriés, mentionnées plus haut dans cette section. Cliquez sur OK une fois que c'est fini.
Cliquez sur Configuration avancée, cochez Utiliser le DNS pour la résolution de noms, et Utiliser LMHOSTS si vous utilisez un fichier de résolution, comme celui mentionné ci-dessus.
Cliquez alors sur OK sur toutes les boites de dialogue, et redémarrez le système.

« Pingez » votre machine Linux pour tester la connexion réseau : Démarrer / Executer, tapez :

```
ping 192.168.1.1
```

(C'est seulement un test de connexion locale, vous ne pouvez pas encore pinger l'extérieur).

Configurer les systèmes UNIX :

Si vous n'avez pas encore installé votre carte réseau et recompilé votre noyau avec le driver adéquat, faites le maintenant.
Installez des outils TCP/IP, comme par exemple le package nettools, si ce n'est déjà fait. J’utilise personnellement l’utilitaire « netcfg » fourni avec la Red HAT.

Affectez IPADDR à 192.168.1.x (1<x<255), puis NETMASK à 255.255.255.0, GATEWAY à 192.168.1.1 et BROADCAST à 192.168.1.255.
Par exemple, sur les systèmes Red Hat Linux, vous pouvez éditer le fichier /etc/sysconfig/network-scripts/ifcfg-eth0, ou simplement le faire par l'intermédiaire du Control Panel.
(c'est différent sur SunOS, BSDi, Slackware Linux, etc...)
Ajoutez l'adresse IP de votre DNS et votre ordre de recherche DNS dans /etc/resolv.conf.
Il sera éventuellement nécessaire de mettre à jout le fichier /etc/networks, selon votre configuration.
Redémarrez les services adéquats, ou, plus simplement, redémarrez votre système.
Testez votre connexion avec la passerelle en utilisant la commande « ping » :

```
ping 192.168.1.1
```

(C'est seulement un test de connexion locale, vous ne pouvez pas encore pinger l'extérieur).

Configuration sous DOS avec le package NCSA :

Si vous n'avez pas encore installé votre carte réseau, faites le maintenant.
Chargez le driver adéquat. Pour une carte NE2000, tapez nwpd 0x60 10 0x300, si votre carte utilise l'IRQ 10 et l'adresse d'entrée/sortie 0x300.
Créez un nouveau répertoire, et décompressez-y l'archive NCSA Telnet : pkunzip tel2308b.zip.
Utilisez un éditeur de texte pour ouvrir le fichier config.tel.
Affectez myip=192.168.1.x (1 < x < 255), et netmask=255.255.255.0.
Dans cet exemple, vous auriez à régler hardware=packet, interrupt=10, ioaddr=60.
Vous devriez avoir au moins une seule machine déclarée comme passerelle, à savoir la machine sous Linux :
name=default
host=le_nom_de_votre_hote_linux hostip=192.168.1.1 gateway=1

Pour mettre en place le DNS :
name=dns.domain.com~; hostip=123.123.123.123; nameserver=1

NB: remplacez les champs par les informations qu'utilise votre machine Linux.
Sauvegardez votre nouveau fichier config.tel.
Lancez un telnet vers la machine Linux pour tester la connexion réseau : telnet 192.168.1.1.

c) Extension : gestion automatique des clients du réseau local

La mise en place d'un serveur DHCP permettrait de s'affranchir de la configuration IP de chaque client du réseau. En effet, les machines configurées de façon à ce qu'elle obtiennent leurs informations TCP/IP automatiquement feront appel à ce serveur DHCP qui leur transmettra une adresse IP, l'adresse IP de la passerelle, du ou des serveurs DNS, …

Pour plus d'informations sur la mise en place d'un serveur DHCP, consultez le manuel « DHCP » du même auteur et de la même édition.

4) Configuration des règles d'IP Forwarding

A ce point du document, vous devriez avoir votre noyau et les autres packages installés, ainsi que les modules nécessaires chargés. De plus, les adresses IP, la passerelle, et le DNS devraient être installés sur les clients du réseau.
Maintenant, la seule chose à faire est d'utiliser l'outil de firewalling IP (ipfwadm ou ipchains) pour faire suivre les paquets appropriés à la machine qui convient.

Personnellement, j'utilise IP Chains 1.3.9. Si vous disposez d'un IP Chains de version inférieures à la 1.3.8, utilisez plutôt IPFwAdm.

Vous allez ainsi déterminer qui (quelles machines) peut utiliser la passerelle pour ce connecter à Internet.
Il faut, pour cela ajouter quelques lignes à la fin du fichier « rc.firewall » précédent :

- pour accorder l'accès total vers l' « extérieur » à tous les utilisateurs du réseau :

```
/sbin/ipchains -P forward DENY
/sbin/ipchains -A forward -s 192.168.1.0/255.255.255.0 -j MASQ
```

- pour limiter l'accès aux seuls machines "steve" (192.168.1.2) et "philippe" (192.168.1.3) :

```
/sbin/ipchains -P forward DENY
# autoriser la machine "steve"
/sbin/ipchains -A forward -s 192.168.1.2/255.255.255.0 -j MASQ
# autoriser la machine "philippe"
/sbin/ipchains -A forward -s 192.168.1.3/255.255.255.0 -j MASQ
# ou : /sbin/ipchains -A forward -s 192.168.1.3/32 -j MASQ
```

Une autre méthode consiste à tout interdire puis à autoriser certains accès :

- Interdiction :

```
# tout refuser
/sbin/ipchains -P input DENY
/sbin/ipchains -P output REJECT
/sbin/ipchains -P forward REJECT
```

- Autorisation :

```
# autoriser l'accès au DNS
# si votre serveur est 192.168.1.1 :
/sbin/ipchains -A input -i ppp0 -p udp -s 192.168.1.1/32 53 -j ACCEPT
/sbin/ipchains -A output -i ppp0 -p udp -d 192.168.1.1/32 53 -j ACCEPT
# ou si vous avez plusieurs serveurs sur le réseau 192.168.1.0 :
/sbin/ipchains -A input -i ppp0 -p udp -s 192.168.1.0/24 53 -j ACCEPT
/sbin/ipchains -A output -i ppp0 -p udp -d 192.168.1.0/24 53 -j ACCEPT
# ppp0 est l'interface pour la connexion Internet, eth0 celle pour le réseau local.
# autoriser la navigation http sur tous les sites :
/sbin/ipchains -A input -i ppp0 -p tcp -s any/0 80 -d any/0 1024:655535 -j ACCEPT
/sbin/ipchains -A output -i ppp0 -p tcp -s any/0 1024:655535 -d any/0 80 -j ACCEPT
```

Détails des commandes et syntaxe IP Chains :

Pour définir les règles de filtrage qui reposent sur trois niveaux de protection (appelés chaînes), input (tout ce qui rentre), output (tout ce qui sort), et forward (tout ce qui est transmis), il faut appliquer, à chaque chaîne de protection, une police DENY, ACCEPT ou REJECT.
Lorsqu'un paquet arrive, ce sont les règles de filtrage de la chaîne input qui vont être appliquées, si le paquet correspond aux règles de cette chaîne, il sera autorisé à continuer sa "route" vers une autre machine, ce sera alors les règles de la chaîne forward qui seront appliquées, si ce paquet doit être réexpédié, ce sera enfin au tour des règles de la chaîne output.
Ce(s) paquet(s) transite(nt) par une interface qu'il est également possible de définir avec l'option -i.
Ces interfaces peuvent etre lo (loopback), eth0 pour la carte du réseau locale ou ppp0 pour le réseau externe (connexion internet PPP).
Il est plus pratique de réaliser un script qui sera lancer depuis /etc/rc.d/rc.local , que d'ajouter les commandes ipchains dans ce fichier rc.local, et de rendre executable ce sript.
Prenons un exemple :

```
# Interdir toutes les entrées
/sbin/ipchains -F input
/sbin/ipchains -P input DENY

#autoriser les postes du réseau local
/sbin/ipchains -A input -s 192.168.1.0/24 -j ACCEPT

# autoriser les connexions dns ftp et http
/sbin/ipchains -A input -i ppp0 -p udp -s 192.168.1.0/24 53 -j ACCEPT
/sbin/ipchains -A input -p tcp -s 192.168.1.0/24 -d 0.0.0.0/0 20 -j ACCEPT
/sbin/ipchains -A input -p tcp -s 192.168.1.0/24 -d 0.0.0.0/0 21 -j ACCEPT
/sbin/ipchains -A input -p tcp -s 192.168.1.0/24 -d 0.0.0.0/0 80 -j ACCEPT

#interdir l'accès au poste 192.168.1.11
/sbin/ipchains -A input -s 192.168.1.11/32 -j REJECT

# Interdir toutes les sorties
/sbin/ipchains -F output
/sbin/ipchains -P output DENY

#autoriser les postes du réseau local
/sbin/ipchains -A output -d 192.168.1.0/24 -j ACCEPT

# autoriser les connexions dns ftp et http
/sbin/ipchains -A output -i ppp0 -p udp -d 192.168.154.0/24 53 -j ACCEPT
/sbin/ipchains -A output -p tcp -s 0.0.0.0/0 20 -d 192.168.154.0/24 -j ACCEPT
/sbin/ipchains -A output -p tcp -s 0.0.0.0/0 21 -d 192.168.154.0/24 -j ACCEPT
/sbin/ipchains -A output -p tcp -s 0.0.0.0/0 80 -d 192.168.154.0/24 -j ACCEPT

#interdir l'accès à l'adresse 195.245.10.55
/sbin/ipchains -A output -d 195.245.10.55/32 -j REJECT

# Interdir la transmission
/sbin/ipchains -F forward
/sbin/ipchains -P forward DENY

# Pour "masquer"(IP_Masquerade) l'adresse des postes du réseau local
# derrière celle de la passerelle: MASQ s'applique à la chaîne Forward
/sbin/ipchains -A forward -s 192.168.1.0/24 -j MASQ
```

DENY refuse tout sans avertir l'émetteur.
ACCEPT accepte tout.
REJECT refuse tout et le signale à l'émetteur.

Les options :
-L pour lister les règles d'une chaîne et la police
-F pour supprimer les règles d'une chaîne mais pas la police
-A pour ajouter une nouvelle règle à une chaîne
-P pour changer, dans une chaîne, la police DENY, ACCEPT ou REJECT
-i interface
-s source
-d destination
-j police
-p protocole

Que ce soit pour l'adresse de la source (-s) ou de la destination (-d), la syntaxe sera :
0.0.0.0/0 pour n'importe quelle adresse
192.168.154.0/24 pour les postes du réseau 192.168.154.0
192.168.154.11/32 pour le poste ayant l'adresse IP 192.168.154.11

Par défaut si -p n'est pas spécifié, la règle concernera tous les protocoles réseaux.

Les protocoles possibles sont udp, tcp et icmp, le fichier /etc/services contient la liste des services réseau avec leur numéro de port ainsi que leur protocole :
20 ftp-data
21 ftp
53 dns
80 www HTTP

Par défaut si -i n'est pas spécifié, la règle concernera tous les interfaces .
De même, si -d n'est pas indiqué, c'est 0.0.0.0/0 qui sera pris en compte .

Pour vérifier les règles de filtrage, voici quelques commandes :

```
/sbin/ipchains -nL
/sbin/ipchains --list
/sbin/ipchains -v -L input
/sbin/ipchains -v -L output
/sbin/ipchains -v -L forward
```

Informations sur IPFwAdm :

Les correspondances entre les commandes IP Chains et IPFwAdm sont disponibles dans le fichier HOWTO d'IP Chains.

Outils graphiques de configuration

Vous pouvez installer l'interface graphique de Daniel Roche, "Easyfw" compatible IP Chains et IPFwAdm, récupérable sur http://www.linux-kheops.com/pub/easyfw/easyfwFR.html

Un autre utilitaire très sympa pour vous aider à configurer votre "FireWall" est GFCC disponible sur http://icarus.autostock.co.kr/

5) Tester la passerelle

Avant tout, vérifier que votre machine passerelle est bien connectée à Internet en ouvrant Netscape et en vous dirigeant sur un site bien connue tel que http://www.redhat.com/.

Ensuite, réinitialisez les règles d'IP Forwarding en tapant les commandes suivantes :

```
[root@redhat rc.d]# /sbin/ipchains -F; /sbin/ipchains -X
```

Ensuite, exécutez votre fichier « rc.firewall ». Vérifiez que les règles ont été chargées :

```
[root@redhat rc.d]# /sbin/ipchains --list
```

Depuis chaque poste client, il suffit d'utiliser la commande « ping », pour vérifier si la connexion vers la passerelle fonctionne, puis pour vérifier si l'on peut traverser cette passerelle en le dirigeant vers une machine bien connue telle que « www.yahoo.com ».

Enfin, testez votre accès avec un browser Internet, puis avec un client FTP (si vous avez chargé le module approprié) vers un ftp public en tentant de télécharger quelques fichiers.

Si tout fonctionne correctement, votre accès est bien configuré.