Configuración de un Gateway

Emiliano Castagnari

3 de junio de 2003

Este documento describe brevemente cómo configurar una máquina corriendo GNU/Linux, como **Gateway**. Esto significa, poder compartir una conexión a Internet desde nuestro GNU/Linux (de ahora en más, **server**) con nuestra red interna, a través del conocido método *masquerade*, una forma particular de NAT (Network Address Translation - Traducción de Direcciones de Red).

Índice

1. Acerca de este documento

1.1. Créditos

- Emiliano Castagnari: Preparación
- Margarita Manterola: Corrección ortográfica. Conversión a Latex.

2. Introducción

Este documento tiene como objetivo ayudar a la gente que recién ingresa al mundo de Linux (y a los que ya están también), a configurar una máquina que esté corriendo un sistema operativo GNU/Linux, que se conecta a Internet ya sea, a través de un ADSL, Cablemodem, o bien una línea telefónica, de forma tal que pueda compartir esa conexión con una red interna, sea en su casa, trabajo, o cualquier otro lugar en el que disponga de una red privada y una conexión a Internet.

2.1. Convenciones del documento

La máquina que dispondrá de la conexión a Internet, sera referida como **server** o **gateway**. Cualquier otra máquina que use la primera para navegar (léase utilice el servicio provisto por el **server**) será denominada **cliente**.

El proveedor de Internet, será referenciado como **ISP** (Internet Service Provider)

2.2. ¿Que es un Gateway?

Un Gateway es un **server**, que proporciona a **clientes** conectividad hacia el mundo exterior, estén o no dentro de una red privada (nosotros nos centraremos en el primer caso, pero para el segundo, esto no cambia demasiado).

Este **server**, puede ser cualquier tipo de máquina, con cualquier sistema operativo que sea capaz de proveer funcionalidades de router y firewall. El documento se basa en "Sistemas Operativos"GNU/Linux, y no en *Sistemas Inoperativos*.

2.3. ¿Por que querría usar uno?

Supongamos la siguiente situación: En tu casa/trabajo disponen de una sola conexión a Internet (léase dirección IP provista por ISP) o bien una sola línea telefónica. A la vez, tenés varias estaciones de trabajo, con diferentes sistemas operativos (o no), desde las cuales se realizan tareas recreativas/laborales que dependen de tu acceso a Internet, y por tener una sola conexión/línea telefónica no pueden estar todas conectadas al mismo tiempo.

Acá es donde toma realmente fuerza el hecho de tener un gateway que proporcione a las demás máquinas pertenecientes a la red, una conexión al mundo exterior.

2.4. Pros y Contras

Pros

- Te podés conectar desde cualquier máquina, o de todas al mismo tiempo!!
- Podés ofrecer servicios a Internet desde distintas maquinas en la red interna
- Tener control de donde se navega, o donde no se puede.

Contras

- Si tenés una conexión de baja velocidad, digamos un acceso telefónico, el ancho de banda será distribuido entre las diferentes máquinas que generen peticiones a Internet. Esto no es tan terrible si vas a navegar y chequear correo, pero si querés jugar algún juego en red con gráficos pesados y demás, vas a tener menor rendimiento.

  Personalmente, creo que es mejor poder conectarte de cualquier máquina sin tener que gritarle a tu hermano que deje de chatear porque le tenés que enviar un TP a un amigo de la facu.

- Tener control de donde se navega, o donde no se puede.

3. Requerimientos

Veremos básicamente tres formas de conectarnos a Internet, lo que determinará la topología de nuestra red.

Digamos que lo ideal es disponer, en el **server**, de dos placas de red, en el caso de tener una conexión ADSL o Cablemodem. Sin embargo, si lo que tenemos es una conexión telefónica, nos será más que suficiente una sola placa de red y el modem.

Nota: Para conectar más de dos máquinas entre sí, es necesario disponer de un Hub o un Switch (centralizadores de conexiones para redes UTP) y utilizar cables UTP rectos/cruzados.

3.1. Hardware

1. Una máquina
2. Para una conexión ADSL, una o dos placas de red y el modem ADSL
3. Para una conexión Cablemodem, dos placas de red
4. Para una conexión Telefónica una placa de red y un modem
5. Un Hub (según el caso)
6. Cables de red (Cruzado o Recto, depende de la topología de red)

3.2. Software

1. Un sistema operativo como la gente: GNU/Linux (cualquier distribución)
2. iptables - iproute (Kernels 2.4), o bien

   ipchains - ifconfig (Kernels 2.2)

4. Elección de rangos IP y un poco de redes

Básicamente, lo que necesitás hacer es elegir un rango de IPs privado. Para la gente que desconoce lo que esto significa, me voy a limitar a contarles que es un rango de IPs que en Internet no van a encontrar nunca, debido justamente a que es "privada", solo las verán, valga la redundancia, en redes privadas.

Estos son algunos de los rangos de IPs privadas disponibles:

Red	Máscara	Descripción
127.0.0.0	255.0.0.0	Dispositivos de loopback (No las podés usar)
10.0.0.0	255.0.0.0	Sin restricciones dentro del rango
172.16.0.0	255.240.0.0(?)	Sin restricciones dentro del rango
192.168.0.0	255.255.0.0	Sin restricciones dentro del rango

Es posible que esto de la red les suene un poco extraño si no están familiarizados con el manejo de los rangos y máscaras, pero no se preocupen, ahora veremos unos ejemplos concretos. Mientras tanto, si les interesa saber como partir una red, ver como funcionan las máscaras y demás, les recomiendo una página útil para comenzar a entender como se disponen estas cosas:

`http://www.htmlweb.net/redes/subredes/subredes_1.html`

Centrándonos un poco más en nuestra red, supongamos que les asignamos un rango de IP del último grupo, es decir, por ejemplo, "192.168.0.0". Nuestra máscara de red será "255.255.255.0". Digamos que nuestra red esta compuesta por un **server** y dos **clientes**, los configuraremos de la siguiente manera:

Máquina	IP	Máscara de red
server	192.168.0.1	255.255.255.0
cliente1	192.168.0.2	255.255.255.0
cliente2	192.168.0.3	255.255.255.0

¿Por qué? Porque así se me ocurrió, nada más que por eso... Igualmente pensá que al seleccionar las IPs tenés que estas dentro del rango xxx.xxx.xxx.1 - xxx.xxx.xxx.254. Si querés saber porqué, leé el documento de la dirección especificada más arriba.

5. Qué es una Topología de red

Por topología de red, se entiende la forma en la cual esta diseñada la red a la que pertenecemos. Esto incluye desde la cantidad de máquinas que hay, hasta la forma en la cual están interconectadas.

6. Preparando un Gateway ADSL

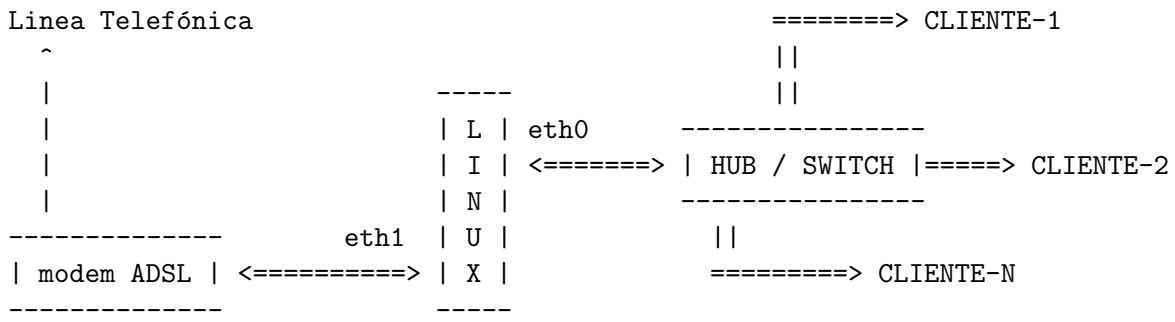
6.1. Situación ideal y Topología de la red

Idealizando la situación, supongamos que tenemos dos placas de red en el **server**, el modem ADSL y dos máquinas **cliente** (o más de estos últimos). En este caso, necesitaremos para conectar toda nuestra red, un Hub o Switch.

Si solamente tenemos el **server** (con dos placas de red), modem ADSL y una sola máquina **cliente**, no será necesario ni un Hub ni un Switch.

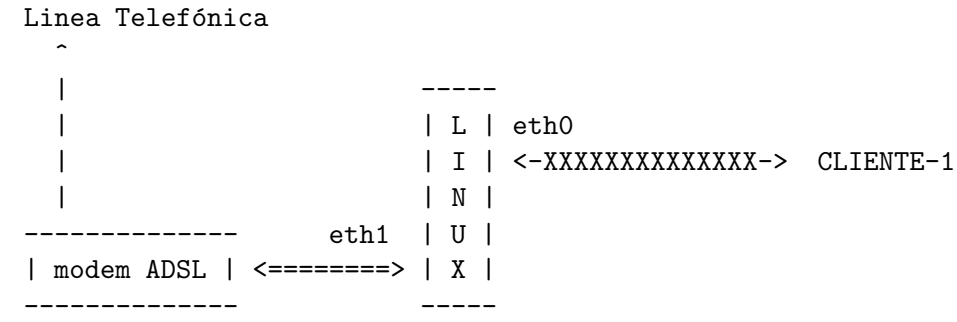
Para el primer caso (dos o más clientes, y un Hub)

El modem ADSL, lo conectamos a una de las placas de red del servidor (digamos eth1), y la otra placa de red (para ser ordenados, eth0) la conectaremos al Hub/Switch, de la siguiente manera:



Para el segundo caso (Un solo cliente, y sin Hub)

Para el modem ADSL no hay cambio, lo conectamos a la placa de red eth1, y la placa restante en el **server**, irá directamente conectada a la placa de red del **cliente**, quedando de la siguiente manera:



Espero se hayan dado cuenta de las diferencias en ambos gráficos, en cuanto al cable UTP utilizado para conectar eth0. Esto no es un error, sino para diferenciar explícitamente el tipo de cable a utilizar, que es muy importante, de lo contrario nuestra red no podrá funcionar.

En el primer caso, el cable UTP que conecta la eth0 del **server** al Hub/Switch y este a los **clientes**, está representado por `<====>`, lo que significa que el tipo de UTP a utilizar es recto"(debido a la topología de red).

Para el segundo ejemplo, el UTP que conecta la eth0 del **server** al **cliente**, está simbolizado por `<-XXXX->`, que nos indica que debemos emplear un UTP *cruzado* (Crossover).

Noten, también, que el cable de conexión desde el modem ADSL a la eth1 del **server** es **SIEMPRE** recto.

Antes de conectarnos a Internet, configuraremos las placas de red del **server** de esta forma (esto se aplica para ambos ejemplos ilustrados anteriormente):

- eth0 : 192.168.0.1/0
- eth1 : 192.168.1.1/32

¿Cómo hacemos esto? Bueno, utilizando iproute, o bien ifconfig. Personalmente prefiero iproute, por ser mas versátil. Una ventaja muy grande sobre ifconfig, es que se puede asignar muy fácilmente una segunda dirección IP a la placa de red.

Necesitamos tener acceso al sistema como superusuario para esto, por lo tanto, si no estas logueado como root, es un buen momento para hacer un `su`.

Nota: ¡¡Todo esto se hace suponiendo que la red no esta configurada!!

Comandos:

```
[user@foo /]$ su
password:
[root@foo /]# ip addr add 192.168.1.1/32 dev eth1
[root@foo /]# ip addr add 192.168.0.1/0 broadcast 192.168.0.255 dev eth0
[root@foo /]# ip addr show

1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:0a:f2:53:90:fe brd ff:ff:ff:ff:ff:ff
   inet 192.168.0.1/0 192.168.0.255 scope global eth0
3: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:1d:44:53:fc:01 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.1/32 scope global eth1
[root@foo /]# logout
[user@foo /]$
```

Lamentablemente, iproute es un paquete muy poco documentado, por lo tanto se aprende rompiendo un poco las cosas. Bueno, en realidad no es tan drástico esto, podes desconfigurar la red, pero romper, no vas a romper nada.

Si querés ver lo que podes hacer con ip, podes por ejemplo, poner "ip addr help".

Brevemente, lo que hicimos fue:

`# ip addr add 192.168.1.1/32 dev eth1`

Le dijimos a IP que agregue (add) una dirección (addr) al dispositivo (dev) eth1. La situación es la misma para eth0, salvo que acá especificamos la dirección de **broadcast** que utilizaremos (para la red /0", siempre es esa).

Luego de haber configurado las dos placas de red, le pedimos a IP que nos muestre (show) las direcciones (addr) de todos los dispositivos de red en el **server**.

Lo más importante, es la siguiente linea:

```
# ip addr show
1: ...
   ...
2: ...
   ...
inet 192.168.0.1/0 brd 192.168.0.255 scope global eth0
```

En donde nos especifica la dirección de la placa, la dirección de **broadcast**, y el dispositivo al que pertenece todo esto (eth0 en este caso)

Una vez hecho esto, la configuración de la red del **server**, está terminada. Nos queda configurar el daemon para conectarse y el firewall.

6.2. Una situación no tan ideal

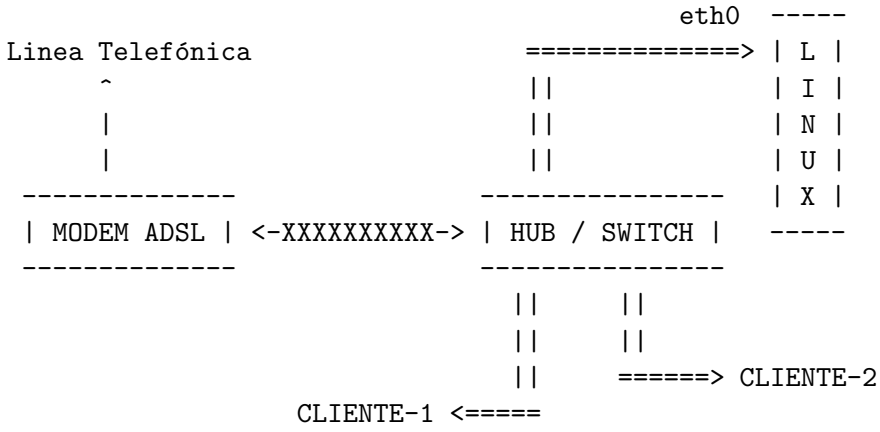
Bien, supongamos ahora que, sea por poco presupuesto, o bien no nos quedan mas slots PCI/ISA, no podemos tener una máquina con dos placas de red, sino una sola.

Por lo tanto, necesitaremos disponer de un **server**, un modem ADSL, un **cliente** (o más) y necesariamente un Hub/Switch. Este último dispositivo, es necesario si queremos armar una

red UTP con estas características.

Nota: Para otro tipo de redes (SLIP, Seriales) no es necesario el Hub.

Veamos entonces, un esquema de nuestra red:



Como pueden ver, la topología de nuestra red cambia radicalmente. Ahora, nuestro modem ADSL se conecta al Hub/Switch (en vez de hacerlo al **server**), y mediante un cable UTP cruzado (en lugar de uno recto).

Diferencias:

Con este tipo de configuración, podés llegar a generarte mas tráfico en el Hub, que también es conocido como un repetidor. Básicamente lo que hace este ultimo, es repetir los paquetes que le llegan de una boca determinada, al las restantes. La placa de red a la cual está destinado este paquete lo acepta, el resto lo descartan.

Esto te va a generar un tráfico bastante alto en el Hub, porque pensá que ahora, los pedidos de los **clientes** se van a hacer al **server** en la interfase eth0, y a la vez este, también va a hacer sus pedidos al modem en esa misma interfaz. El modem a su vez, responderá a esta última, generando así una congestión muy alta en el repetidor, y que como consecuencia, provocara colisiones entre paquetes, y posibles perdidas de estos.

Si tenés un Switch, todo este tema va a ser manejado mucho mejor por el dispositivo, que no es propiamente dicho un repetidor. Muy por encima de su funcionamiento real (porque tampoco lo conozco con exactitud), les puedo contar que posee un caché que le permite recordar en qué puertos están conectadas las máquinas a las que le corresponde el paquete que está recibiendo o enviando, reduciendo radicalmente el tráfico en él.

Actualmente, el precio de un Hub ronda los U$S 50 acá en Argentina, y un Switch, bueno, mejor no hablar de eso. Si estas buscando la opción económica, te propongo un Hub o comprarte otra placa de red si tenés slots libres.

Volviendo a lo que es la configuración, nos resta configurar la placa de red eth0 del **server**, de la siguiente forma:

```
[root@foo /]# ip addr add 192.168.0.1/0 dev eth0
```

Si querés saber qué hicimos, leé la sección anterior.

7. Preparando un Gateway Cablemodem

... TODO ...

8. Preparando un Gateway Telefónico

... TODO ...

9. Configurando el Daemon de Conexión y un Firewall

Bien, en este momento nos encontramos con la configuración de red de nuestro **servidor** terminada. Lo siguiente que haremos, será configurar los daemons necesarios para cada tipo de conexión, y a su vez, configurar un firewall que nos permita proveer Internet a nuestra red interna.

9.1. Acceso ADSL

Deberemos tener instalados los protocolos ppp y pppoe, ambos disponibles en cualquier distro, y normalmente instalados por defecto. Para configurar el daemon de conexión de ADSL, hay distintas herramientas que nos posibilitan una configuración rápida y efectiva. Haremos mención de dos de ellas, las mas utilizadas (y las únicas que conozco), que son el rp-pppoe, y el pppoeconf.

El pppoeconf viene empaquetado por defecto en la distribución Debian Woody, pero de no haberlo instalado, pueden simplemente hacer:

```
[root@foo /]# apt-get install pppoeconf
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  pppoeconf
0 packages upgraded, 1 newly installed, 0 to remove and 0  not upgraded.
Need to get 15.9kB of archives. After unpacking 201kB will be used.
Get:1 http://ftp.br.debian.org stable/main pppoeconf 0.9.10.6 [15.9kB]
```

Para cualquier otra distribución, esta disponible el paquete *r*p-pppoe, que puede ser descargado en su forma binaria(rpm's), o su código fuente. No tiene mayores complicaciones al compilarlo, y pueden obtener ambos de la siguiente dirección:

`http://www.roaringpenguin.com/pppoe/` en donde pueden además, encontrar su documentación online, instalación, y demás cosas ...

9.1.1. pppoeconf

... TODO ...

9.1.2. rp-pppoe

Una vez compilado, o instalado, es sumamente fácil utilizarlo. Primero que nada, deberemos configurar varios parámetros que nos permitirán acceder a Internet, y esto lo haremos usando un script de configuración llamado **adsl-setup**.

Este nos preguntara las siguientes cosas:

- Nombre de usuario
- Interfaz en la cual esta conectado el Modem ADSL
- Activar la conexión bajo demanda
- Información sobre DNS's

- Contraseña
- Activar Firewall
- Confirmación de datos

- Nombre de Usuario

  Deberemos ingresar el nombre de usuario que nos fue asignado por el **ISP**

- Interfaz en la cual esta conectado el Modem ADSL

  ... es bastante autoexplicativo ... ¿no?

- Activar conexión bajo demanda

  En caso que no queramos que el **servidor** este conectado permanentemente, deberemos seleccionar esta opción. Esto hará que el enlace establecido con el **ISP** sea cortado si no hay actividad después de cierto tiempo. Lo más recomendable, es, como estamos utilizando una conexión que no tiene costo alguno, que el enlace este activo, por mas que no se registre actividad alguna, esto no nos generara ningún tipo de problemas.

- Información sobre DNS's

  De activar esta opción, nuestro archivo `/etc/resolv.conf` será modificado por el daemon, utilizando así, los DNS's provistos por el *D*HCP de nuestro **servidor**.

- Contraseña

  Se nos solicitara ingresar la contraseña ... 2 veces.

- Activar Firewall

  Acá podremos seleccionar 3 tipos diferentes de firewall, desde ninguno, hasta un símil de paranoia.

  Les recomiendo que no seleccionen ninguno, ya que lo configuraremos nosotros antes de conectaremos.

- Confirmación de Datos

  Terminada la configuración, deberemos decidir si esta todo bien.

Con esto, estamos listos para conectarnos, ejecutando el comando **adsl-start**. No obstante, no nos convendría hacerlo todavía, debido a que nuestro *f*irewall todavía no esta corriendo, y estaríamos exponiéndonos a la intrusión a nuestra red de algún usuario malintencionado.

Por lo tanto, el próximo paso, es configurar nuestro *f*irewall !!

9.2. Acceso Cablemodem

... TODO ...

9.3. Acceso Telefónico

... TODO ...

10. Configurando el Firewall y compartiendo la conexión

10.1. Un poco de teoría

Primero que nada, un poco de teoría básica sobre lo que realmente esta pasando al compartir una conexión a Internet con una red interna (no necesitas leerlo, pero digamos que "lo que no te mata, te hace mas fuerte").

El método empleado que utilizaremos para compartir nuestra conexión, es el conocido como *M*ASQUERADE. Este, es una forma particular de *N*AT, *N*etwork Address Translation (Traducción de Direcciones de Red), denominado *S*OURCE NAT o también referido por sus siglas *S*NAT.

El escenario es el siguiente, por un lado tenemos la red de redes, y nuestra red privada. Por como están dispuestas las cosas (digamos, dogmáticamente), las IPs que pertenecen a nuestra red privada, no pueden tener contacto alguno con las direcciones publicas de internet. Es por esto que necesitamos nuestro **gateway** para comunicarnos con el resto del mundo exterior.

¿Pero cómo nos comunicamos realmente? Bueno, de esto se encarga el *S*OURCE NAT. Nuestros pedidos por una pagina web, consulta POP, SMTP, y cualquier otro tipo de servicio en la red publica, son realizados a la interfase privada de nuestro **server**, y este identifica que el pedido es para el mundo publico de internet. Pero surge el inconveniente que nuestra IP es privada, por lo tanto no puede existir en la red de redes. En consecuencia nuestro **gateway** modifica el pedido que ha realizado el **cliente** cambiándole la *I*P fuente (IP del **cliente**) por su IP (IP del **server**), y marcando el paquete de forma tal que, cuando la respuesta remota es recibida, es identificada por nuestro *f*irewall y redireccionada.[a]l **cliente** que origino el pedido.

Por lo tanto, esto significa que los pedidos que se realizan en la red interna, hacia la red publica, serán vistos por esta ultima como si hubiesen sido originados por la interfaz publica del **server**, dicho de otra forma, para Internet, el origen de los paquetes enviados desde la red interna sera, *S*IEMPRE, la IP publica asignada por nuestro **ISP** al **server**.

Si querés saber mas sobre este tema, fijate en la "Bibliografía recomendada para el hambre de conocimiento". 8-D

10.2. Construcción del Firewall, ensuciándonos las mandos...

Nuestro primer *f*irewall sera bastante rudimentario, pero lo suficientemente bueno como para poder controlar quien utiliza nuestro **gateway** para acceder a la Nebulosa Internet, y lo haremos de forma tal, que podrá ser reutilizado cualquiera sea nuestro tipo de conexión.

```
#!/bin/bash
####================================================####
##
## Firewall v0.1
##
## Autor: Emiliano Castagnari - Jueves 13, Marzo 2003
##
## Licencia: Este script es provisto bajo
##           licencia GNU/GPL.
##
##
##
```

```
## Script para la configuración de un firewall hogareño
##
## Dudas y/o comentarios, mandalos a la lista del
## LugFI (http://www.fi.uba.ar/lug)
##
####==============================================####


## Interfaz publica,
##
##  ADSL / Telefónica => PUB_IF="ppp0"
##  Cablemodem        => PUB_IF="eth0"
##
##  Para los usuarios de Cablemodem, la interfaz
##  publica, es aquella donde entra el cable de red
##  proveniente del splitter (divisor) del coaxil

PUB_IF="ppp0"


## Interfaz Privada
## Esta es la interfaz que forma parte de nuestra red interna
## y que tiene el mismo rango de IP que los clientes

PRIV_IF="eth0"


## Red Privada
## Esta, es la red privada a la cual pertenecemos, por ejemplo
## 192.168.0.0/0

NET="192.168.0.0/0"

#################################################################
##### FIN DE PARÁMETROS CONFIGURABLES ... ####################
#################################################################

## La siguiente sección depende de ciertos paquetes instalados
## en tu distribución, podes modificar lo que sigue a gusto
## siempre y cuando sepas como hacerlo 8-).
##
#### Algunos programas que utilizaremos ... #################
IPTABLES="$(which iptables)"
MODPROBE="$(which modprobe)"
DEPMOD="$(which depmod)"
LSMOD="$(which lsmod)"
GREP="$(which grep)"
AWK="$(which awk)"
CAT="$(which cat)"
```

```
SED="$(which sed)"
IP="$(which ip)"
################################################################

## IP Publica
## Esta es la IP que se nos asigno al conectarnos mediante la
## interfaz $PUB_IF

PUB_IP="$($IP addr show dev $PUB_IF|$AWK '/inet/ { print $2 }')"


## IP Privada
## Esta es la IP privada de nuestro gateway/servidor asignada
## a la interfaz $PRIV_IF

PRIV_IP="$($IP addr show dev $PRIV_IF|$AWK '/inet/ { print $2 }')"


################################################################
## Necesitas ser root para esto !!

if [ "$UID" != "0" ]; then

 $CAT << _MSG

  ---------------------------------------
  | Atención !!                         |
  | Necesitas acceso como super usuario |
  | para ejecutar iptables !!!!!!!!!!!! |
  |                                     |
  | ... No se cargara el firewall !!    |
  ---------------------------------------

_MSG

 exit 0

fi

## Si ip me devuelve el rango de red, se lo saco
## tanto en la PUB_IP, como en PRIV_IP

chk=$(echo $PUB_IP|$GREP "\/[0-9][0-9]$");
if [ -n "$chk" ]; then
 PUB_IP=$(echo $PUB_IP|$SED "s/\/[0-9][0-9]$//")
fi

chk=$(echo $PRIV_IP|$GREP "\/[0-9][0-9]$");
if [ -n "$chk" ]; then
```

```
 PRIV_IP=$(echo $PRIV_IP|$SED "s/\/[0-9][0-9]$//")
fi


## Muestro un pequeño esquema de como esta
## nuestra conf actual de red

$CAT << _MSG

 ..... [ Cargando Firewall ] .....
 .
 · Interfaz Publica : ..... $PUB_IF
 · Interfaz Privada : ..... $PRIV_IF
 .
 · IP Publica : ..... $PUB_IP
 · IP Privada : ..... $PRIV_IP
 · Red Privada : ..... $NET
 .
 · Cargando Módulos ...
 .
 .....
_MSG


############################################################
## Comienzo la carga de módulos para iptables. ###########
## Primero corroboro las dependencias de los
## módulos del kernel ...

$DEPMOD -a

## Cargo los módulos, verificando que no estén ya cargados
##
## Los módulos cargados son los siguientes:
##
## · ip_tables
## · iptable_nat
## · ip_nat_ftp
## · ip_conntrack
## · ip_conntrack_ftp
## · ip_conntrack_irc
##


for modulo in ip_tables iptable_nat ip_nat_ftp ip_conntrack \
              ip_conntrack_ftp ip_conntrack_irc; do

 if [ -z "$($LSMOD |$GREP $modulo)" ]; then
```

```
  echo " · cargando $modulo"
  $MODPROBE $modulo

 fi

done

#### Finalización de carga de Módulos ###################
############################################################

echo " · Módulos Cargados ..."

echo " · Habilitando forwarding ..."
echo 1 > /proc/sys/net/ipv4/ip_forward

echo " · Políticas por defecto eliminadas ..."

############################################################
#### Acá comienza la magia de iptables ...

## Reglas por defecto, muy estrictas.

## · Permito la entrada de paquetes, y reinicio
##   la cadena que me lo permite (INPUT)
## · Permito la salida de paquetes, y reinicio
##   la cadena que lo permite (OUTPUT)
## · Deniego por defecto todos los paquetes
##   que me pidan ser redireccionados y reinicio
##   esa cadena (FORWARD)
## · Reinicio la tabla de NATeo

$IPTABLES -P INPUT ACCEPT
$IPTABLES -F INPUT
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -F OUTPUT
$IPTABLES -P FORWARD DROP
$IPTABLES -F FORWARD
$IPTABLES -t nat -F


## Sección de LOGS
##  Si deseo loguear la actividad de la red por el syslog
##  descomentar las siguientes lineas.
##  NOTA: si utilizas mucho la red, tus logs pueden crecer mucho
##        controla el espacio ocupado cada tanto

## Logueo los paquetes que salen del server generados por las
## maquinas de la red, en los que estoy haciendo SNAT.
$IPTABLES -t nat -A POSTROUTING --src $NET -o $PUB_IF -j LOG \
```

```
          --log-prefix 'NATTING : '

## !!! Alguien que no pertenece a la red está usando nuestro server ...
$IPTABLES -t nat -A POSTROUTING --src ! $NET -o $PUB_IF -j LOG \
          --log-prefix 'INTRUSION !! : '

## Logueo el trafico desde el exterior hacia adentro, que no tengan como
## puerto de origen el 22 (SSH) - solo para ilustrar -
$IPTABLES -A FORWARD -i $PUB_IF -o $PRIV_IF -p tcp --sport ! 22 \
          -j LOG --log-prefix 'FWD IN: '

## Logueo todos los paquetes que entren por la interfase publica y
## cuya fuente es cualquier IP sobre internet, dirigida al puerto 22
## (SSH) - útil para saber si alguien esta conectado -
$IPTABLES -A INPUT -i $PUB_IF --src 0/0 -p tcp --dport 22 -j LOG \
          --log-prefix 'CONEXION SSH: '

#########################################################################

## Sección SOURCE NAT
## En esta parte, llevamos a cabo la vedette de todos esto,
## el SNAT, quien nos permite compartir la conexión

## Solo voy a permitir que los paquetes entren desde afuera hacia la
## red interna cuando el estado de esta conexión haya sido establecida
## desde adentro, y este activa - ESTABLISHED, RELATED -
$IPTABLES -A FORWARD -i $PUB_IF -o $PRIV_IF -m state \
          --state ESTABLISHED,RELATED -j ACCEPT

## Voy a permitir todo el trafico de la red interna hacia afuera, siempre
## y cuando pertenezca a la red la IP que origina el paquete ... sirve
## para proteger en cierta forma del spoofing, por eso, si nosotros
## administramos la red, debemos mantenerla lo mas chica posible.
$IPTABLES -A FORWARD -i $PRIV_IF --src $NET -o $PUB_IF --dst 0/0 -j ACCEPT

## Habilito finalmente el SOURCE NAT, otra vez, para la red a la que
## pertenecemos
$IPTABLES -t nat -A POSTROUTING --src $NET -o $PUB_IF -j MASQUERADE

#########################################################################


echo " ·····[ Firewall Cargado ]·····"

--- /CUT ---
```