Instalación y configuración del cortafuegos

Sergio González González

Universidad de León, España

sergio.gonzalez@hispalinux.es

Este documento es una pequeña guía que muestra cómo instalar y configurar el cortafuegos desarrollado para el proyecto. El proceso está especificado para una distribución Debian GNU/Linux.

Introducción

El cortafuegos utilizado ha sido desarrollado íntegramente por el grupo GSO a partir de información y otros cortafuegos disponibles en Internet. Está programado en el lenguaje shell de *bash* y se distribuye bajo la licencia libre GNU/GPL.

Utiliza las características avanzadas de los kernels de Linux de la serie 2.4.x para el filtrado de paquetes de red. La herramienta utilizada es **iptables**, que está englobada en el proyecto Netfilter (http://www.netfilter.org/).

Las características más importantes del cortafuegos desarrollado son:

- Filtrado de paquetes por estado
- Enmascaramiento del tráfico local hacia Internet
- Redirección transparente de puertos (tanto en el tráfico de entrada como de salida)

Mediante este cortafuegos, se da acceso a internet a todos los ordenadores de la Unidad de Imagen, así como a todos los ordenadores situados en el laboratorio F1.

Instalación

Necesitamos descargar el archivo del cortafuegos, que está situado en la sección de recursos en la página web del proyecto. El nombre completo es "rc_firewall".

Una vez lo hayamos descargado, lo copiaremos en su ubicación correcta, que es /etc/init.d/ mediante el comando:

```
# cp rc_firewall /etc/init.d/
```

Una vez copiado, necesitamos añadirlo a los scripts de arranque para que se ejecute automáticamente cada vez arranque el ordenador. Para ello, ejecutaremos:

```
# update-rc.d rc_firewall defaults
```

Una vez finalizado el proceso, podremos proceder a la configuración de los parámetros.

Si no añadimos el cortafuegos a los scripts de arranque, deberemos ejecutarlo manualmente cada vez que inicie el ordenador.

En vez de descargar el cortafuegos de la página web (o si ésta no se encuentra disponible en esos momentos), se puede copiar el código de la la sección de nombre *Firewall* en un archivo con el mismo nombre y realizar el proceso con dicho archivo.

Configuración

La configuración del cortafuegos se realiza mediante la edición del archivo /etc/init.d/rc_firewall.

A continuación, veremos las variables más importantes que debemos tener en cuenta para una correcta configuración y funcionamiento del cortafuegos:

Configuración básica

El rango de IP en el que están los ordenadores de nuestra red:

```
RANGO_IP_LAN_CLIENTES="192.168.2.0/24"
```

Las direcciones de broadcast de la red local y de la IP pública:

```
DIRECCION_BCAST_LAN_CLIENTES="192.168.2.255/32"
DIRECCION_BCAST_INET="193.146.99.255/32"
```

El nombre de la tarjeta de red conectada a internet:

```
IFACE_INET="eth0"
```

El nombre de la tarjeta de red conectada a la red local:

```
IFACE_LAN_CLIENTES="eth1"
```

Necesitamos indicar que el cortafuegos hará de router:

```
ROUTER="yes"
```

La dirección IP pública:

```
NAT="193.146.99.249"
```

También debemos indicar qué servicios queremos que sean accesibles desde el exterior. Podemos indicar el nombre o el número de puerto correspondiente:

```
SERVICES_TCP="ftp ssh 110 http 443 3128 2401 9999 143"
```

Configuración avanzada

El desarrollo del cortafuegos ha estado fuertemente condicionado por las características concretas (filtrados y redirecciones de puertos muy específicas) de las redes en las que iba a funcionar, con lo que no es adaptable trivialmente a otras situaciones distintas. En esos casos, es necesaria una modificación más avanzada del código, que debido a la orientación de esta guía, no se tratará.

Para dichas situaciones, se recomienda la lectura de la información recomendada en la la sección de nombre *Más información* antes de hacer las adptaciones necesarias al código expuesto en la la sección de nombre *Firewall*.

Firewall

Este es el fichero completo del cortafuegos. Se han respetado los comentarios originales para tener una mejor compresión del significado de algunas variables y reglas:

```
#!/bin/sh
#
# rc.firewall - Script de inicio para el cortafuegos incluído
#               en el núcleo Linux 2.4.x
#
# Todas las líneas que comiencen con # son comentarios y no
# se tendrán en cuenta.
#
# Este script está basado en los scripts incluídos en los
# tutoriales de:
#
#  - Oskar Andreasson  <blueflux@koffein.net>
#
#        (http://people.unix-fu.org/andreasson/),
#
#  - Daniel Robbins <drobbins@gentoo.org>
#
#        (http://ibm.com/develworks)
#
# El presente script ha sido readaptado por:
#
#  - Sergio González González <sergio.gonzalez@hispalinux.es>
#  - Luis Llorente Campo <luis.llorente@hispalinux.es>
#
#
# Este script es software libre; lo puedes redistribuir y/o
# modificar bajo los términos de la Licencia Pública General GNU.
# (http://www.gnu.org)
#
# This script is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License.
# (http://www.gnu.org)
#
#
#                     #######################
```

```
#                     # Comienzo del script #
#                     #######################
#


##
# Definimos una función encargada de limpiar las reglas creadas
# anteriormente en Iptables.
#
# reset_and_flush() restablece Iptables con los valores por defecto.
#

reset_and_flush ()
{

    echo -en "\033[47m\033[34m                                                                        \033[0m\n"
    echo -en "\033[47m\033[34m   * Limpiando Iptables...                                     \033[0m"

    IPTABLES=/sbin/iptables

    # Restablecemos la política por defecto de la tabla de filtrado.

      ${IPTABLES} -P INPUT ACCEPT
      ${IPTABLES} -P FORWARD ACCEPT
      ${IPTABLES} -P OUTPUT ACCEPT

    # Restablecemos la política por defecto de la tabla nat.

      ${IPTABLES} -t nat -P PREROUTING ACCEPT
      ${IPTABLES} -t nat -P POSTROUTING ACCEPT
      ${IPTABLES} -t nat -P OUTPUT ACCEPT

    # Restablecemos la política por defecto de la tabla mangle.

      ${IPTABLES} -t mangle -P PREROUTING ACCEPT
      ${IPTABLES} -t mangle -P OUTPUT ACCEPT

    # Limpiamos todas las reglas existentes en las tablas de filtrado, nat y mangle.

      ${IPTABLES} -F
      ${IPTABLES} -t nat -F
      ${IPTABLES} -t mangle -F

    # Borramos todas las cadenas que no están por defecto en las tablas de filtrado,
    # nat y mangle.

      ${IPTABLES} -X
      ${IPTABLES} -t nat -X
      ${IPTABLES} -t mangle -X

    echo -en "\033[47m\033[1;32m[done]   \033[0m\n"
}
```

```
##
# Analizamos el parámetro pasado. Los posibles valores son:
#
#  - start -> Activamos las reglas del cortafuegos listadas
#             en este script.
#
#  - stop  -> Eliminamos todas las reglas del cortafuegos,
#             dejando la política por defecto del mismo en
#             "allow".
#
#  - Si no se han pasado argumentos, se informa del uso del script.
#


case "$1" in


#################
# SIN ARGUMENTOS
#
# No se han pasado argumentos...

 "") echo -en "Uso: rc_firewall {start|stop}\n"

    ;; # Fin de ""


########
# START
#
# Se ha pasado el argumento "start"

 "start")


    #######################################################
    # Datos relativos a la red en la que nos encontramos: #
    #######################################################


    ##
    #    - RANGO_IP_LAN -> Muestra la red local
    #      (/24 significa que sólo usaremos los 24 primeros bits de
    #       los 32 bits que forman una dirección IP. Es lo mismo que
    #       la máscara de subred 255.255.255.0).
    #

      RANGO_IP_LAN_CLIENTES="192.168.2.0/24"
      RANGO_IP_LAN_DMZ="192.168.3.0/24"
      RANGO_IP_LAN_PROYECTOS="192.168.4.0/24"
```

```
##
#     - IP_LOCALHOST -> Dirección IP del localhost.
#

  IP_LOCALHOST="127.0.0.1/32"


##
#     - IP_LAN  -> Indica cual es la IP del localhost en la red
#                  local.

  IP_LAN_CLIENTES="192.168.2.1/32"
  IP_LAN_DMZ="192.168.3.1/32"
  IP_LAN_PROYECTOS="192.168.4.1/32"

##
#     - IP_INET -> Informa de la IP externa que posee el "cortafuegos"
#                  (en caso de ser fija). El uso de esta variable puede
#                  ser un riesgo de seguridad, pero algunas veces es lo
#                  que quiero. Si no tienes una IP stática, te sugiero
#                  que no uses esta opción.

  IP_INET="193.146.99.249/32"


##
#     - IP_DINAMICA -> Si poseemos una IP asignada dinámicamente, puedes
#                      descomentar la siguiente línea, y se encargará
#                      de obtener la IP.

#  IP_DINAMICA=`/sbin/ifconfig eth1 | grep 'inet addr' | awk '{print $2}' | awk -F: '{print $2}'`


##
#     - DIRECCION_BCAST_LAN -> Contiene la dirección broadcast de
#                              la red local.
#

  DIRECCION_BCAST_LAN_CLIENTES="192.168.2.255/32"
  DIRECCION_BCAST_LAN_DMZ="192.168.3.255/32"
  DIRECCION_BCAST_LAN_PROYECTOS="192.168.4.255/32"
  DIRECCION_BCAST_INET="193.146.99.255/32"

##
#     - IFACE_INET -> informa de la tarjeta de red conectada a internet.

  IFACE_INET="eth0"


##
#     - IFACE_LAN -> Informa de la tarjeta de red conectada a la red local.
```

```
  IFACE_LAN_CLIENTES="eth1"
  IFACE_LAN_DMZ="eth3"
  IFACE_LAN_PROYECTOS="eth2"

##
#    - IFACE_LO  -> Informa del interface localhost

  IFACE_LO="lo"

##
#    - IPTABLES -> Indica la ruta en la que podemos encontrar el programa
#                  iptables.

  IPTABLES="/sbin/iptables"


##
#    - ROUTER -> Si necesitas actuar como un router (y así poder pasar
#                paquetes IP entre dos tarjetas de red), necesitas
#                la asignación ROUTER="yes"; Si este no es tu caso, has de
#                poner ROUTER="no"
#

  ROUTER="yes"


##
# Cambia la siguiente línea por la dirección, direcciones o rango de direcciones
# IPs estáticas que poseas para hacer SNAT estático. Si tienes una IP dinámica,
# establece el valor "dynamic". Si no necesitas ningún tipo de NAT
# (Network Address Translation), establece NAT como "" para desactivarla.
#

  NAT="193.146.99.249"

##
# Cambia la siguiente línea de forma que liste todas las interfaces de red que tienes,
# incluyendo lo.

  INTERFACES="lo eth0 eth1 eth2 eth3"


##
# Cambia la siguiente línea de forma que liste los números asignados o los nombres
# simbólicos (de /etc/services) de todos los servicios que quieras dar al público
# en general. Si no quieres dar ningún servicio, establece el valor como "".
#
#   SERVICES_TCP -> Servicios que se van a dar por los puertos tcp.
#
#   SERVICES_UDP -> Servicios que se van a dar por los puertos udp.
#
#   SERVICES_ICMP -> Servicios que se van a dar por los puertos icmp.
#
```

```
  ##
  # TCP:
  #
  #      ftp-data    -> 20
  #      ftp         -> 21
  #      ssh         -> 22
  #      http        -> 80
  #      auth        -> 113
  #      netbios-ns  -> 137
  #      netbios-dgm -> 138
  #      netbios-ssn -> 139
  #      cvspserver  -> 2401
  #
  # UDP:
  #
  #      domain      -> 53
  #      netbios-ns  -> 137
  #      netbios-dgm -> 138
  #      netbios-ssn -> 139
  #      icpv2       -> 3130
  #
  # ICMP:
  #
  #      Echo Reply              -> 0
  #      Destination Unreachable -> 3
  #      Redirect                -> 5
  #      Echo Request            -> 8
  #      Time Exceeded           -> 11
  #
  # (Para más información sobre ICMP mirar:
  #  http://www.ee.siue.edu/~rwalden/networking/icmp.html
  #  ftp://sunsite.unc.edu/pub/docs/rfc/rfc792.txt)
  #


    SERVICES_TCP="ftp ssh 110 http 443 3128 2401 9999 143"
    SERVICES_ICMP="0 3 5 8 11"

####################
# Carga de módulos #
####################


  echo -en "\n"
  echo -en "\033[47m\033[31m                                                                      \033[0m\n"
  echo -en "\033[47m\033[31m                     Configurando el cortafuegos                      \033[0m\n"
  echo -en "\033[47m\033[31m                                                                      \033[0m\n"
  echo -en "\033[47m\033[31m                                                                      \033[0m\n"

##
# Cargamos todos los módulos necesarios de IPTables.
#
```

```
# La siguiente línea es necesaria para inicializar la
# carga de módulos.

  echo -en "\033[47m\033[31m                                                                        \033[0m\n"
  echo -en "\033[47m\033[34m   * Cargando los modulos...                                  \033[0m"

  /sbin/depmod -a

##
# Adds some iptables targets like LOG, REJECT and MASQUARADE.
# /sbin/modprobe ipt_LOG
# /sbin/modprobe ipt_REJECT
# /sbin/modprobe ipt_MASQUERADE
#
# Support for owner matching


  /sbin/modprobe ipt_owner


##
# Support for connection tracking of FTP and IRC.
#

  /sbin/modprobe ip_conntrack_ftp
  /sbin/modprobe ip_nat_ftp
# /sbin/modprobe ip_conntrack_irc


  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"




###################################################
# Restablecemos el estado por defecto de Iptables #
###################################################

  reset_and_flush



###############################################
# Aplicando distintas protecciones y opciones #
###############################################


##
# Route verification is where a packet which comes from an unexpected
# interface is dropped: for example, if your internal network has
# addresses 10.1.1.0/24, and a packet with that source address comes
# in your external interface, it will be dropped
#
```

```
# Protección contra IP spoofing

  echo -en "\033[47m\033[31m                                                                        \033[0m\n"
  echo -en "\033[47m\033[34m   * Cargando las reglas:                                               \033[0m\n"
  echo -en "\033[47m\033[34m                                                                        \033[0m\n"
  echo -en "\033[47m\033[30m       Desactivando spoofing en todas las interfaces...       \033[0m"

#disable spoofing on all interfaces

  for x in ${INTERFACES}
  do
     echo 1 > /proc/sys/net/ipv4/conf/$x/rp_filter
  done

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"

# Deshabilitamos el ECN (explicit congestion notification) si ha sido
# compilado en el núcleo.
# El porqué de esta acción es que no todas las redes lo soportan
# de momento y puede dar problemas de conexión.

  echo -en "\033[47m\033[30m       Desactivando el ECN...                                 \033[0m"

  if [ -e /proc/sys/net/ipv4/tcp_ecn ]
  then
      echo 0 > /proc/sys/net/ipv4/tcp_ecn
  fi

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"

# SYNCOOKIES

  echo -en "\033[47m\033[30m       Activando protección contra syncookie...               \033[0m"

  echo 1 > /proc/sys/net/ipv4/tcp_syncookies

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"



######################################################
# Definiendo la política por defecto del cortafuegos #
######################################################


##
# Cadenas INPUT, OUTPUT y FORDWARD
#


  echo -en "\033[47m\033[34m                                                                        \033[0m\n"
  echo -en "\033[47m\033[34m       - Políticas por defecto para el cortafuegos:                     \033[0m\n"
  echo -en "\033[47m\033[34m                                                                        \033[0m\n"
```

```
  echo -en "\033[47m\033[30m            INPUT   -> DROP...                              \033[0m"

  $IPTABLES -P INPUT DROP

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"
  echo -en "\033[47m\033[30m            OUTPUT  -> DROP...                              \033[0m"

  $IPTABLES -P OUTPUT DROP

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"
  echo -en "\033[47m\033[30m            FORWARD -> DROP...                              \033[0m"

  # $IPTABLES -P FORWARD DROP

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"

# $IPTABLES -t mangle -P PREROUTING DROP
# $IPTABLES -t mangle -P OUTPUT DROP
# $IPTABLES -t nat -P PREROUTING DROP
# $IPTABLES -t nat -P POSTROUTING DROP
# $IPTABLES -t nat -P OUTPUT DROP


############################################
# Creando nuevas cadenas en el cortafuegos #
############################################

##
# Creamos cadenas separadas para ICMP, TCP y UDP.
#

  echo -en "\033[47m\033[34m                                                                     \033[0m\n"
  echo -en "\033[47m\033[34m       - Creación de nuevas cadenas para el cortafuegos:               \033[0m\n"
  echo -en "\033[47m\033[34m                                                                     \033[0m\n"

  echo -en "\033[47m\033[30m            ICMP    -> icmp_packets...                      \033[0m"

  $IPTABLES -N icmp_packets

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"

  echo -en "\033[47m\033[30m            TCP     -> tcp_packets...                       \033[0m"

  $IPTABLES -N tcp_packets

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"

  echo -en "\033[47m\033[30m            UDP     -> udpincoming_packets...               \033[0m"

  $IPTABLES -N udpincoming_packets

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"
```

```
##
# Cadena allowed
#

  echo -en "\033[47m\033[30m            ALLOWED -> allowed...                             \033[0m"

  $IPTABLES -N allowed

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"


#######################################################
# Definición de las reglas para las distintas cadenas #
#######################################################


##
# Reglas para ALLOWED
#

  echo -en "\033[47m\033[34m                                                                         \033[0m\n"
  echo -en "\033[47m\033[34m       - Definición de reglas para las cadenas:                          \033[0m\n"
  echo -en "\033[47m\033[34m                                                                         \033[0m\n"

  echo -en "\033[47m\033[30m            Definiendo reglas para ALLOWED...                 \033[0m"

  # Esta cadena será utilizada si alguien intenta conectar con un puerto "allowed"
  # desde Internet. Si está abriendo una conexión, o si ya tenía una establecida
  # ACPETAREMOS el paquete, si no lo denegamos. Aquí es donde la correspondencia
  # de estado entra en juego, permitimos los paquetes cuyas conexiones ya estén
  # establecidas (ESTABLISHED) y relaccionadas (RELATED).
  #

  $IPTABLES -A allowed -p TCP --syn -j ACCEPT
  $IPTABLES -A allowed -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT

  # Hacemos que nuestro cortafuegos responda a las peticiones en los puertos
  # TCP y UDP, indicando que no existe ningún servicio disponible en el puesto
  # solicitado. Con esto evitamos que un atacante sepa que estamos
  # detrás de un cortafuegos, haciéndole ver que no disponemos de ningún
  # servicio en nuestro sistema, de esta forma igual se va a por otro
  # equipo, y deja de molestar ;-)

  #      $IPTABLES -A allowed -p TCP -i $IFACE_INET -j REJECT --reject-with tcp-reset

  # Si la línea anterior no está comentada, esta regla nunca se alcanza,
  # pero la dejo sin comentar. Lo que hace la siguiente regla es
  # descartar (sin devolver respuesta) todos los paquetes TCP que le lleguen

  $IPTABLES -A allowed -p TCP -j DROP

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"
```

```
##
# Reglas para ICMP
#
# Se abrirán los servicios listados en SERVICES_ICMP
#

  echo -en "\033[47m\033[30m            Definiendo reglas para ICMP...                    \033[0m"

  for x in ${SERVICES_ICMP}
  do
      ${IPTABLES} -A icmp_packets -p ICMP -s 0/0 --icmp-type ${x} -j ACCEPT
  done

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"



##
# Reglas para TCP
#
# Se abrirán los servicios listados en SERVICES_TCP

  echo -en "\033[47m\033[30m            Definiendo reglas para TCP...                     \033[0m"

  # ${IPTABLES} -A INPUT -p tcp ! --syn -m state --state NEW -j LOG --log-prefix "New not syn:"
  # ${IPTABLES} -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

  for x in ${SERVICES_TCP}
  do
     ${IPTABLES} -A tcp_packets -p TCP -s 0/0 --dport ${x} -j ACCEPT
  done


  ##
  # Reject auth
  #

  # $IPTABLES -A tcp_packets -p TCP -i $IFACE_INET --dport 113 -j REJECT

  ##
  # Reject Xms Scans
  #
  #
  # Generic dirty interface maping

  # $IPTABLES -A tcp_packets -p TCP --tcp-flags ALL FIN,URG,PSH -j LOG --log-level DEBUG -m limit --limit 1/s
  # $IPTABLES -A tcp_packets -p TCP --tcp-flags ALL FIN,URG,PSH -j DROP

  ##
  # Reject Fin scans
```

```
  #

  # $IPTABLES -A tcp_packets -p TCP --tcp-flags ALL FIN -m state --state ! ESTABLISHED -j LOG --log-level DEBUG -m limit --limit 1/s
  # $IPTABLES -A tcp_packets -p TCP --tcp-flags ALL FIN -m state --state ! ESTABLISHED -j DROP

  ##
  # Reject ANY station that opens and immediately closes a connection
  # Some portscanners does this
  #

  # $IPTABLES -A tcp_packets -p TCP --tcp-flags ALL SYN,FIN -j LOG --log-level DEBUG -m limit --limit 1/s
  # $IPTABLES -A tcp_packets -p TCP --tcp-flags ALL SYN,FIN -j DROP


  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"



##
# Reglas para UDP
#

  echo -en "\033[47m\033[30m            Definiendo reglas para UDP...                     \033[0m"

  for x in ${SERVICES_UDP}
  do
       $IPTABLES -A udpincoming_packets -p UDP -s 0/0 --source-port ${x} -j ACCEPT
  done


  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"



##
# Reglas para FORWARD
#

  echo -en "\033[47m\033[30m            Definiendo reglas para FORWARD...                 \033[0m"

  $IPTABLES -A FORWARD -p tcp ! --syn -m state --state NEW -j LOG --log-prefix "New not syn:"
  $IPTABLES -A FORWARD -p tcp ! --syn -m state --state NEW -j DROP

  $IPTABLES -A FORWARD -i $IFACE_LAN_CLIENTES -j ACCEPT
  $IPTABLES -A FORWARD -i $IFACE_LAN_PROYECTOS -j ACCEPT


  # PROYECTOS

  # $IPTABLES -A FORWARD -p tcp --dport 21  -i $IFACE_LAN_PROYECTOS -j ACCEPT
  # $IPTABLES -A FORWARD -p tcp --dport 22  -i $IFACE_LAN_PROYECTOS -j ACCEPT
  # $IPTABLES -A FORWARD -p tcp --dport 80  -i $IFACE_LAN_PROYECTOS -j ACCEPT
  # $IPTABLES -A FORWARD -p tcp --dport 999 -i $IFACE_LAN_PROYECTOS -j ACCEPT
```

```
  $IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT


  # Syn-flood protection:

  # $IPTABLES -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT

  # Furtive port scanner:

  # $IPTABLES -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j ACCEPT

  # Ping of death:

  #  $IPTABLES -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT


  $IPTABLES -A FORWARD -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level DEBUG --log-prefix "IPT FORWARD packet died: "

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"



##
# Reglas para PREROUTING
#
# Hacemos un chequeo para bloquear falsas IP's obvias.

  echo -en "\033[47m\033[30m            Definiendo reglas para PREROUTING...              \033[0m"

  # $IPTABLES -t nat -A PREROUTING -i $IFACE_INET -s 192.168.0.0/16 -j DROP
  # $IPTABLES -t nat -A PREROUTING -i $IFACE_INET -s 10.0.0.0/8 -j DROP
  # $IPTABLES -t nat -A PREROUTING -i $IFACE_INET -s 172.16.0.0/12 -j DROP
  # $IPTABLES -t nat -A PREROUTING -i $IFACE_LAN_PROYECTOS ! -s 192.168.4.0/16 -j DROP
  # $IPTABLES -t nat -A PREROUTING -i $INET_IFACE -s $IP_INET -j DROP



  # $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 80 \
  #                                                      -j DNAT --to-destination 192.168.2.2
  # $IPTABLES -t nat -A PREROUTING -p TCP -i $INET_IFACE -d $IP_INET --dport 21 \
  #                                                      -j DNAT --to-destination 192.168.2.2
  # $IPTABLES -t nat -A PREROUTING -p TCP -i $INET_IFACE -d $IP_INET --dport 22 \
  #                                                      -j DNAT --to-destination 192.168.2.2

  ##
  # Reject Xms Scans
  #
  #
  # This disallows ALL portscans that will hit the PREROUTING table

  # $IPTABLES -t nat -A PREROUTING -p tcp --tcp-flags ALL FIN,URG,PSH -j LOG --log-level DEBUG -m limit --limit 1/s
  # $IPTABLES -t nat -A PREROUTING -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
```

```
  ##
  # Reject Fin scans
  #
  #
  # This disallows ALL portscans that will hit the PREROUTING table

  # $IPTABLES -t nat -A PREROUTING -p tcp --tcp-flags ALL FIN -j LOG --log-level DEBUG -m limit --limit 1/s
  # $IPTABLES -t nat -A PREROUTING -p tcp --tcp-flags ALL FIN -j DROP

  ##
  # Reject ANY station that opens and immediately closes a connection
  # Some portscanners does this
  #

  # $IPTABLES -t nat -A PREROUTING -p tcp --tcp-flags ALL SYN,FIN -j LOG --log-level DEBUG -m limit --limit 1/s
  # $IPTABLES -t nat -A PREROUTING -p tcp --tcp-flags ALL SYN,FIN -j DROP

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"



##
# Reglas para INPUT
#
# establish the basic INPUT chain and filter the packets onto the correct
# chains.

   #
   # Take care of bad TCP  packets that we don't want
   #

    $IPTABLES -A INPUT -p tcp ! --syn -m state --state NEW -j LOG \
                            --log-prefix "New not syn:"
    $IPTABLES -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

    $IPTABLES -A INPUT -p ICMP -j icmp_packets
    $IPTABLES -A INPUT -p TCP  -j tcp_packets
    $IPTABLES -A INPUT -p UDP  -j udpincoming_packets
    $IPTABLES -A INPUT -p ALL -i ${IFACE_LAN_CLIENTES} -d ${DIRECCION_BCAST_LAN_CLIENTES} -j ACCEPT
    $IPTABLES -A INPUT -p ALL -i ${IFACE_LAN_PROYECTOS} -d ${DIRECCION_BCAST_LAN_PROYECTOS} -j ACCEPT
    $IPTABLES -A INPUT -p ALL -i ${IFACE_LO} -d $IP_LOCALHOST -j ACCEPT
    $IPTABLES -A INPUT -p ALL -d $IP_LAN_CLIENTES -j ACCEPT
    $IPTABLES -A INPUT -p ALL -d $IP_LAN_PROYECTOS -j ACCEPT
    $IPTABLES -A INPUT -p ALL -d $IP_INET -m state --state ESTABLISHED,RELATED -j ACCEPT
    $IPTABLES -A INPUT -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level DEBUG --log-prefix "IPT INPUT packet died: "
    # ${IPTABLES} -A INPUT -p TCP -i ${IFACE_INET} -j REJECT --reject-with tcp-reset
    # ${IPTABLES} -A INPUT -p UDP -i ${IFACE_INET} -j REJECT --reject-with icmp-port-unreachable




##
```

```
 # OUTPUT chain
 #
 # establish the basic OUTPUT chain and filter them onto the correct chain


   $IPTABLES -A OUTPUT -p tcp ! --syn -m state --state NEW -j LOG --log-prefix "New not syn:"
   $IPTABLES -A OUTPUT -p tcp ! --syn -m state --state NEW -j DROP

   $IPTABLES -A OUTPUT -p ALL -s $IP_LOCALHOST -j ACCEPT
   $IPTABLES -A OUTPUT -p ALL -s $IP_LAN_CLIENTES -j ACCEPT
   $IPTABLES -A OUTPUT -p ALL -s $IP_LAN_PROYECTOS -j ACCEPT
   $IPTABLES -A OUTPUT -p ALL -s $IP_INET -j ACCEPT
   $IPTABLES -A OUTPUT -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level DEBUG --log-prefix "IPT OUTPUT packet died: "




 ##
 # MANGLE chain
 #
 #
 # invalid crap
 #

 # $IPTABLES -t mangle -A PREROUTING -j LOG --log-level DEBUG -m state --state INVALID -m limit --limit 1/s




 #################################################
 # Reglas para NAT (Network Address Translation) #
 #################################################


 # Enable simple IP FORWARDing and Masquerading
 #
 # NOTE: The following is an example for an internal LAN, where the lan
 # runs on eth1, and the Internet is on eth0.
 #
 # Please change the network devices to match your own configuration

   if [ "$ROUTER" = "yes" ]
   then

echo -en "\033[47m\033[34m                                                                      \033[0m\n"
echo -en "\033[47m\033[34m      - Reglas para el ROUTER:                                        \033[0m\n"
echo -en "\033[47m\033[34m                                                                      \033[0m\n"
echo -en "\033[47m\033[30m           Activando IP forwarding...                       \033[0m"

#we're a router of some kind, enable IP forwarding

  echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
        echo -en "\033[47m\033[1;32m[done]   \033[0m\n"

if [ "$NAT" = "dynamic" ]
        then

            #dynamic IP address, use masquerading

            # Dynamic IP users:
    #
    # If you get your IP address dynamically from SLIP, PPP, or DHCP, enable this
    # option. This enables dynamic-ip address hacking in IP MASQ, making the connection
    # with Diald and similar programs much easier.

      echo -en "\033[47m\033[30m            Activando dynamic-ip address hacking...           \033[0m"

      echo "1" > /proc/sys/net/ipv4/ip_dynaddr

      echo -en "\033[47m\033[1;32m[done]   \033[0m\n"

      echo -en "\033[47m\033[30m            Activando ip-masquerading...                      \033[0m"

              $IPTABLES -t nat -A POSTROUTING -o $IFACE_INET -j MASQUERADE

      echo -en "\033[47m\033[1;32m[done]   \033[0m\n"


elif [ "$NAT" != "" ]
        then
            #static IP, use SNAT

              echo -en "\033[47m\033[30m            Activando SNAT (IP estática)...                   \033[0m"

              iptables -t nat -A POSTROUTING -o $IFACE_INET -j SNAT --to-source ${NAT}

              echo -en "\033[47m\033[1;32m[done]   \033[0m\n"

        fi
  fi


 ##
 # Redirección del tráfico web saliente a la caché de squid.
 # Esto se hace de forma transparente al usuario.

 echo -en "\033[47m\033[30m                                                                     \033[0m\n"
 echo -en "\033[47m\033[30m       Activando proxy transparente...                        \033[0m"

  $IPTABLES -t nat -A PREROUTING -i $IFACE_LAN_CLIENTES -p tcp --dport 80 -j REDIRECT --to-port 3128
  # $IPTABLES -t nat -A PREROUTING -i $IFACE_LAN_DMZ -p tcp --dport 80 -j REDIRECT --to-port 3128
  # $IPTABLES -t nat -A PREROUTING -i $IFACE_LAN_PROYECTOS -p tcp --dport 80 -j REDIRECT --to-port 3128

 echo -en "\033[47m\033[1;32m[done]   \033[0m\n"
```

```
##
# Redirección del tráfico web entrante hacia cancerbero.

  echo -en "\033[47m\033[30m       Activando la redirección del tráfico web a litio...    \033[0m"

  $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 80 -j DNAT --to-destination 193.146.99.248:8080
  $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 443 -j DNAT --to-destination 193.146.99.248:443

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"


##
# Redirección del tráfico web entrante hacia litio.

  echo -en "\033[47m\033[30m       Activando la redirección del tráfico web a litio...    \033[0m"

  $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 110 -j DNAT --to-destination 192.168.2.2:80

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"


##
# Redirección del tráfico apt-proxy entrante hacia litio.

  echo -en "\033[47m\033[30m       Activando la redirección del tráfico apt-proxy a litio...    \033[0m"

  $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 9999 -j DNAT --to-destination 192.168.2.2:9999

  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"


##
# Redirección del tráfico ftp entrante hacia litio.

  echo -en "\033[47m\033[30m       Activando la redirección del tráfico ftp a litio...    \033[0m"

  $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 21 -j DNAT --to-destination 192.168.2.2:21
  $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 20 -j DNAT --to-destination 192.168.2.2:20


  echo -en "\033[47m\033[1;32m[done]   \033[0m\n"


##
# Redirección del tráfico ssh entrante hacia litio.

   echo -en "\033[47m\033[30m       Activando la redirección del tráfico ssh a litio...    \033[0m"

   $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 22 -j DNAT --to-destination 192.168.2.2:22

   echo -en "\033[47m\033[1;32m[done]   \033[0m\n"
```

```
    ##
    # Redirección del tráfico ssh entrante por el puerto 2222 hacia potasio.

       echo -en "\033[47m\033[30m       Activando la redirección del tráfico ssh a potasio...    \033[0m"

       $IPTABLES -t nat -A PREROUTING -p TCP -i $IFACE_INET -d $IP_INET --dport 2222 -j DNAT --to-destination 192.168.4.2:22

       echo -en "\033[47m\033[1;32m[done]   \033[0m\n"


    echo -en "\033[47m\033[31m                                                                      \033[0m\n"
    echo -en "\033[47m\033[31m                                                                      \033[0m\n"
    echo -en "\033[47m\033[31m                      Cortafuegos configurado                         \033[m\n"
    echo -en "\033[47m\033[31m                                                                      \033[0m\n"

   ;; # Fin de "start"



#######
# STOP
#
# Se ha pasado el argumento "stop"

 "stop")

   ##
   # Restablecemos los valores por defecto de Iptables
   #

   echo -en "\n"
   echo -en "\033[47m\033[31m                                                                      \033[0m\n"
   echo -en "\033[47m\033[31m                      Parando el cortafuegos                          \033[0m\n"
   echo -en "\033[47m\033[31m                                                                      \033[0m\n"

   reset_and_flush

   echo -en "\033[47m\033[31m                                                                      \033[0m\n"
   echo -en "\033[47m\033[31m                                                                      \033[0m\n"
   echo -en "\033[47m\033[31m                        Cortafuegos parado                            \033[0m\n"
   echo -en "\033[47m\033[31m                                                                      \033[0m\n"

   ;; # Fin de "Stop"

esac
```

Prueba

Una vez esté todo configurado, pondremos en marcha el cortafuegos mediante el comando:

```
# /etc/init.d/rc_firewall start
```

A partir de este momento, podremos acceder a Internet con cualquiera de los ordenadores clientes. Para comprobarlo, podemos intentar alcanzar a una página web como por ejemplo:

```
# ping http://www.google.com
```

También podemos probar desde otra red si funciona el acceso a los servicios que se han definido (ftp, web, ssh, ...).

Si se quiere aislar la red temporalmente por cualquier motivo, podemos parar el cortafuegos mediante el comando:

```
# /etc/init.d/firewall stop
```

Más información

Se recomienda la lectura de la documentación de **iptables**, que está situada en el directorio /usr/share/doc/iptables/.

Otra documentación muy interesante se encuentra en el sitio web del proyecto Netfilter: http://www.netfilter.org (http://www.netfilter.org/). En ella podremos encontrar numerosos tutoriales sobre **iptables** que nos ayudarán en su compresión y manejo, así como en la teoría de redes necesaria para comprender muchos de los conceptos utilizados en el cortafuegos:

- Netfilter Hacking HOWTO
- Netfilter Extensions HOWTO
- Packet Filtering HOWTO
- Networking Concepts HOWTO
- NAT HOWTO

Uno de los tutoriales más completos es el situado en http://iptables-tutorial.haringstad.com, cuya lectura es más que recomendable.

Sobre este documento

Este documento ha sido escrito en formato XML utilizando la DTD de DocBook (http://www.docbook.org). Mediante este sistema, puede ser fácilmente transformado a múltiples formatos (HTML, TXT, PDF, PostScript, LaTeX, DVI, ...). Se recomienda su utilización como herramienta de documentación potente y libre.