Diald Como

Andrés Seco *AndresSH@ctv.es* v1.0, 15 de Febrero de 2000

Este documento trata de mostrar varios escenarios típicos de utilización de *Diald* para facilitar su puesta en marcha. Los escenarios presentados varían desde una conexión para uso personal en un equipo conectado mediante módem y PPP a un proveedor de servicios de internet (ISP), de manera que no haya que utilizar scripts del tipo `pon/poff` o `ppp-on/ppp-off`, hasta un servidor proxy/firewall con diferentes ISPs para acceder a internet.

Índice General

1 Introducción

1.1 Objetivos

Este documento trata de mostrar varios escenarios típicos de utilización de *Diald* para facilitar su puesta en marcha.

Los escenarios presentados varían desde una conexión para uso personal en un equipo conectado mediante módem y PPP a un proveedor de servicios de internet (ISP), de manera que no haya que utilizar scripts del tipo `pon/poff` o `ppp-on/ppp-off`, hasta un servidor proxy/firewall con diferentes ISPs para acceder a internet.

En el presente documento se tratan los siguientes escenarios:

- Conexión mediante módem y PPP de un equipo aislado a un proveedor de servicios de internet (ISP).
- Conexión mediante módem y PPP de un equipo aislado a varios proveedores de servicios de internet (ISPs) no simultáneamente.
- Conexión mediante módem y PPP de un proxy/firewall a un proveedor de servicios de internet (ISP).

En sucesivas versiones de este documento se cubrirán otros escenarios, como múltiples instancias de *Diald* en ejecución, uso de lineas RDSI y lineas usadas tanto para generar llamadas como para recibirlas.

Anteriormente, existía un Diald-mini-Como realizado por Harish Pillay *h.pillay@ieee.org*, que presentaba una conexión a un ISP usando autentificación basada en chat (login y password previos al arranque de pppd, sin usar PAP ni CHAP).

En el documento se incluirán ficheros de configuración de ejemplo que servirán para la puesta en marcha. Para obtener el máximo rendimiento y usar todas las características de los programas que se utilizarán será necesario leer la documentación adjunta a cada programa y reconfigurar dichos ficheros de configuración que se incluyen como ejemplos.

Así mismo, los directorios en los que se sitúan los ficheros de configuración pueden variar dependiendo de la distribución Linux que se use. Si en la distribución de Linux que usted usa algún fichero se encuentra en directorio diferente al indicado, por favor, hágamelo saber para incluir una nota al respecto.

1.2 Nuevas versiones

La ultima versión de este documento puede ser encontrada en mi pagina web *http://www.ctv.es/USERS/andressh/linux*, en formato SGML y HTML. Otras versiones y formatos pueden ser encontrados en castellano en el Insflug, *http://www.insflug.org/documentos/Diald-Como/*, y en otros idiomas en el LDP - Proyecto de Documentación de Linux, *http://www.linuxdoc.org*.

1.3 Agradecimientos

Quiero dar las gracias a las personas que me ayudaron a poner en marcha mi primer Diald con sus ficheros de ejemplo (alguien de quien no me acuerdo como se llamaba, Mr Cornish Rex y Hoo Kok Mun), a los que me han aportado sugerencias y correcciones a este documento, a los futuros traductores de este documento a otros idiomas y sobre todo a la gente que ha desarrollado y desarrolla *Diald* para los que lo usamos.

2 Copyright y renuncia de responsabilidad

Por supuesto, estoy abierto a todo tipo de sugerencias y correcciones sobre el contenido de este documento.

3 Descripción rápida del funcionamiento de Diald

En pocas palabras, *Diald* lo que hace es generar un interfaz de red nuevo y establecerlo como ruta por defecto. Este interfaz no es real (en la documentación original lo llaman `proxy interface`). *Diald* monitoriza este interfaz y cuando llegan paquetes hacia él, lanza la conexión `ppp`, espera a que se establezca y cambia la dirección de la ruta por defecto sobre la nueva interface `ppp` (normalmente `ppp0`).

Diald controla qué paquetes han llegado al interface y de qué tipo para decidir si los considera a efectos de lanzar la conexión `ppp`, mantenerla activa, hacerla caer o no hacer nada, y en caso de que hayan sido paquetes para tener en cuenta, cuánto tiempo debe estar levantado el interface después de que haya llegado el paquete.

Finalmente, si no hay más tráfico y se acaba el tiempo de espera que el último de los paquetes enviados estableció, *Diald* finaliza la conexión.

Asimismo, es posible controlar los días y horas a los que se puede conectar y los que no, de modo que se puedan aprovechar tarifas bajas u horarios de bajo tráfico.

Hasta aquí, la descripción es válida tanto para la versión 0.16.5 de *Diald* como para las posteriores, pero las versiones posteriores a esta también incluyen funcionalidades adicionales, tales como lista de permisos para decidir a qué usuarios se aplican las normas que deciden si el enlace se levanta, se mantiene o se tira, contabilización avanzada de paquetes transmitidos, mejor integración con lineas RDSI, mejora de rendimiento en base a usar como interfaz proxy un dispositivo `ethertap` (algo así como un interfaz de red que lee/escribe sobre un socket en vez de una tarjeta de red física) en vez de `slip`, conexiones de backup y algunas otras funcionalidades.

4 Notas sobre la identificación en el momento de la conexión

Siempre que nos conectemos a un proveedor de acceso a internet, será necesario enviar un nombre de usuario y una contraseña. Para ello hay diversas posibilidades que nos vendrán impuestas por el proveedor.

Además de las 3 opciones expuestas, también es posible que la conexión no necesite autentificación (normalmente cuando el destino de la llamada es otro ordenador también nuestro y no deseamos autentificar).

4.1 Usuario y contraseña - Login y password

Actualmente no es un método muy utilizado en entornos de acceso a internet.

La identificación se realiza antes de lanzar el demonio `pppd`, y es el propio programa marcador, normalmente `chat`, el que envía los identificadores de usuario y clave. Este envío de usuario y clave se realiza «en claro», por lo que no puede ser considerado seguro.

Un ejemplo de script para el programa `chat` en el que sea necesario indicar usuario y contraseña antes de lanzar `pppd` podría ser el siguiente:

```
ABORT BUSY
ABORT "NO CARRIER"
ABORT VOICE
ABORT "NO DIALTONE"
ABORT "NO ANSWER"
"" ATZ
OK ATDT_NumeroDeTelefono_
CONNECT \d\c
ogin _NombreUsuario_
assword _Contraseña_
```

Las 2 ultimas lineas son las que definen el nombre del usuario y la contraseña y cuando deben ser enviadas (después de recibir «ogin» y «assword» respectivamente. No se suelen poner las primeras letras de dichas palabras para no preocuparse de que unos servidores las envíen en mayúsculas y otros en minúsculas).

Este script suele estar en un fichero en el directorio `/etc/chatscripts`, y, suponiendo que el fichero de script se llama `provider`, puede ser llamado con la linea siguiente:

```
/usr/sbin/chat -v -f /etc/chatscripts/provider
```

4.2 PAP - Password Authentication Protocol

En caso de que el servidor de acceso al que nos conectamos requiera PAP como protocolo para realizar la autentificación de nuestra conexión, durante el establecimiento de la sesión LCP de PPP se negociará dicho protocolo, es decir, una vez establecida la conexión de `chat` y lanzado `pppd`, será este quien realice el envio del nombre de usuario y clave, buscando en el fichero `/etc/ppp/pap-secrets` los valores que debe usar. Este fichero tiene acceso de lectura y escritura solo para `root`, de modo que nadie que no sea el administrador vea su contenido con las claves.

PAP es un protocolo poco seguro, ya que envía la clave sin cifrar, como texto libre, por lo que puede ser leída por alguien que analice la linea de transmisión.

Ejemplo de contenido de /etc/ppp/pap-secrets:

```
_NombreUsuario_ * _Contraseña_
```

4.3 CHAP - Challenge Authentication Protocol

En caso de que el servidor de acceso al que nos conectamos requiera CHAP como protocolo para realizar la autentificación de nuestra conexión, durante el establecimiento de la sesión LCP de PPP se negociará dicho protocolo, es decir, una vez establecida la conexión de chat y lanzado pppd, será este quien realice el envio del nombre de usuario y clave, buscando en el fichero /etc/ppp/chap-secrets los valores que debe usar. Este fichero tiene acceso de lectura y escritura solo para root, de modo que nadie que no sea el administrador vea su contenido con las claves.

CHAP es un protocolo más seguro que PAP, ya que nunca se envía la clave por la linea de transmisión sin cifrar. En su lugar, el equipo que realiza la autentificación envía un identificador aleatorio con el que codificar la clave en el lado del cliente para después enviar esta clave codificada.

Ejemplo de contenido de /etc/ppp/chap-secrets:

```
_NombreUsuario_ * _Contraseña_
```

Es posible que un proveedor de acceso en ocasiones utilice PAP y en otras ocasiones utilice CHAP, por lo que es común definir en ambos sitios el usuario y la clave a utilizar.

5 Notas sobre la resolución de nombres DNS

En todas las conexiones a un ISP es necesario configurar la resolución de nombres DNS, de modo que nuestro ordenador pueda encontrar las direcciones IP asociadas a un nombre de ordenador en internet.

Las direcciones IP de los servidores encargados de realizar esta traducción de direcciones se encuentran en el fichero /etc/resolv.conf.

Lo habitual en un equipo aislado que accede a internet es que este fichero contenga las direcciones IP de los servidores DNS que el proveedor indique:

```
#Fichero /etc/resolv.conf para acceso a internet mediante nombreISP
nameserver 111.222.333.444
nameserver 222.333.444.555
```

En los equipos que realizan tareas de firewall/proxy, es habitual que el fichero contenga la dirección del propio servidor (o la dirección del interfaz loopback, 127.0.0.1) y este además incluya un servidor de nombres DNS que a su vez realizará las traducciones de nombres a direcciones IP poniéndose en contacto con otros servidores DNS.

```
#Fichero /etc/resolv.conf para resolución local de nombres DNS
nameserver 127.0.0.1
```

La puesta en marcha de un servidor DNS está fuera del alcance de este documento. Existe mucha documentación al respecto, pero una aproximación rápida puede obtenerse en el DNS-Howto (*http://www.linuxdoc.org/HOWTO/DNS-HOWTO.html*) cuenta con una traducción en *http://www.insflug.org/documentos/DNS-Como/*.

6 Conexión mediante módem y PPP de un puesto aislado a un ISP

De forma esquemática, serán necesarios los siguientes pasos previos:

- Tener montado el paquete *Diald*. Lo más rápido será montar el que la distribución de Linux que usemos incluya.
- Configurar la resolución de nombres DNS (fichero `/etc/resolv.conf`).
- Comprobar que es posible establecer una conexión con el ISP. Si la distribución incluye utilidades para configurar una conexión, será más rápido usarlas (pppconfig en Debian, kppp si usas KDE, etc). Si tiene problemas para establecer la conexión, los documentos PPP-Howto (*http://www.linuxdoc.org/HOWTO/PPP-HOWTO.html*), Modem-Howto (*http://www.linuxdoc.org/HOWTO/Modem-HOWTO.html*) y Serial-Howto (*http://www.linuxdoc.org/HOWTO/Serial-HOWTO.html*) pueden resultar de ayuda. Ver sección 10 (Más información) si busca traducciones de dichos documentos.
- Establecer los nombres de usuario y contraseña en los ficheros `/etc/ppp/pap-secrets` y `/etc/ppp/chap-secrets`, como se comentó anteriormente.

Y ya entrando definitivamente en *Diald*:

- Preparar el fichero de configuración de *Diald* (`/etc/diald/diald.options` para la versión 0.16.5 de *Diald* y `/etc/diald/diald.conf` para las demás).
- Preparar el fichero de filtros `/etc/diald/standard.filter`, o mejor, dejar ese como está y modificar una copia suya que podemos llamar `/etc/diald/personal.filter`.
- Preparar archivo de comandos para realizar la llamada (`/etc/diald/diald.connect` con permiso de ejecución) y fichero de instrucciones para `chat` (`/etc/chatscripts/provider`), que será utilizado por el anterior.
- Preparar los archivos de comandos `/etc/diald/ip-up` y `/etc/diald/ip-down` si se quieren utilizar (deben tener permiso de ejecución).
- Preparar los archivos de establecimiento de rutas `/etc/diald/addroute` y `/etc/diald/delroute` si se quieren utilizar (deben tener permiso de ejecución). Este paso no suele ser necesario cuando solo se usa una instancia de *Diald*.
- Finalmente, lanzar el demonio `diald` («`/etc/init.d/diald start`» para Debian, «`/etc/rc.d/init.d/diald start`» en RedHat). Normalmente, la instalación del paquete *Diald* habrá generado los scripts necesarios para que arranque cuando se enciende el equipo en los directorios /etc/rcX.d que corresponda.

Si se realiza algún cambio en el fichero de configuración de *Diald* posteriormente a su lanzamiento, deberá ser relanzado («`/etc/init.d/diald restart`» en Debian, «`/etc/rc.d/init.d/diald restart`» en RedHat).

6.1 Fichero `/etc/diald/diald.options` o `diald.conf`

En este fichero de ejemplo que se muestra hay que revisar:

- Puerto en donde se encuentra conectado el módem. Opción `device`.
- Velocidad del módem. Opción `speed`.

- Nombre del usuario para la conexión ppp. Opción `pppd-options`.
- Temporizadores de reintentos.
- Horario de conexión habilitada. Opciones `restrict`.
- Decidir si se usan los archivos de comandos `ip-up` e `ip-down`. Opciones `ip-up` e `ip-down`.
- Decidir si se usan los archivos de comandos `addroute` y `delroute`. Opciones `addroute` y `delroute`. Normalmente solo se usan para configuraciones de enrutamiento complejas o varias instancias de *Diald* en ejecución simultanea.
- Decidir si se usan los filtros estándar o personales. Opciones `include`.

```
##########################
# /etc/diald/diald.options

# Dispositivo donde esta conectado el módem
device /dev/ttyS0

# Archivo de log de diald
accounting-log /var/log/diald.log

# Cola de monitorización de diald
#fifo /var/run/diald/diald.fifo

# Depuración activada al máximo
# Activar la depuración disminuye el rendimiento
#debug 31

# Usamos PPP sobre la linea de transmisión
mode ppp

# Direcciona IP de mi equipo (al conectar se modifica automáticamente con
# la IP asignada por el ISP)
local 127.0.0.5

# Direccion IP del equipo remoto (al conectar se modifica automáticamente
# con la IP del equipo que recibe nuestra llamada)
remote 127.0.0.4

# Mascara de subred
netmask 255.255.255.0

# Indica que las direcciones IP serán asignadas en el momento del
# establecimiento de la conexión
dynamic

# Si la conexión se cae, restablecerla solo si hay trafico de salida
two-way

# Cuando la conexión esta establecida, cambiar las rutas para que apunten
# al interface real, en vez de al interface proxy. No hacer esto implica
# una perdida de rendimiento de un 20 por ciento. Hay kernels antiguos que
# no soportan reroute. Ver el manual de diald para mas información
reroute
```

```
# Diald deberá establecer la ruta por defecto al interface SLIP usado como
# proxy
defaultroute

# Script para establecer rutas personalizadas
#addroute "/etc/diald/addroute"
#delroute "/etc/diald/delroute"

# Scripts para ejecutar cuando el enlace se activa o se desactiva.
# En las versiones 0.9x de Diald hay una opción llamada ip-goingdown para
# ser lanzado al finalizar pero cuando todavía está activo el enlace.
ip-up /etc/diald/ip-up
#ip-down /etc/diald/ip-down

# Scripts usados para establecer la comunicación y pararla
connect "/etc/diald/diald.connect"
#disconnect "/etc/diald/diald.disconnect"

# Usar bloqueo UUCP
#lock

# Conectamos a través de módem. ATENCIÓN: No especificar estas opciones
# en las opciones para PPP, ya que entrarían en conflicto. Para ver las
# opciones de PPP que no se pueden usar en pppd-options ver la pagina
# de manual de diald y buscar pppd-options
modem
crtscts
speed 115200

# Algunos temporizadores
# para información detallada, ver manual de Diald
connect-timeout 120
redial-timeout 60
start-pppd-timeout 120
died-retry-count 0
redial-backoff-start 4
redial-backoff-limit 300
dial-fail-limit 10

# Opciones para pasar a pppd
# Estas opciones también se pueden poner en /etc/ppp/options, que son las
# opciones por defecto de pppd, pero si es necesario tener diferentes
# configuraciones de diald es necesario ponerlas aquí
# noauth - no autentificar al destino de la llamada.
#          "Infovía Plus" (España) no se identifica frente a nuestro
#          equipo.
# user - poner aquí el usuario y el isp separados por la arroba (@).
pppd-options noauth user usuario@isp

# Restricciones de horario de uso.
# Esta sección debe estar antes de los filtros.
# El comando restrict es experimental y puede cambiar en futuras versiones
# de diald (esta sintaxis es para la versión 0.16)
# Solo usar en horario nocturno para plan
```

```
# Bononet Noche (España-Telefónica)
# Lunes a Viernes de 0 a 8 y de 18 a 24
# Domingos de 0 a 24
restrict 8:00:00 18:00:00 1-5 * *
down
restrict * * * * *

# Sin condiciones especiales de tarificación
# (primeros segundos cobrados de una vez, unidad de tarificación en
# segundos, tiempo en segundos para comprobar desconexión)
#impulse 0,0,0
# Bononet Noche (España-Telefónica) tarifica por segundos a partir del 160
impulse 160,0,0
# si se tarificase por minutos y cobrasen los 10 primeros de una vez:
#impulse 600,60,10

# Filtros estándar
#include /etc/diald/standard.filter
# o Filtros personales
include /etc/diald/personal.filter
```

6.2 Fichero de filtros personal

La manipulación de este fichero debe ser realizada de forma muy cuidadosa. Es en este fichero en el que se decide cuándo y porqué se trata de levantar el enlace, mantenerlo, tirarlo o ignorarlo, dependiendo del tipo de tráfico.

En general, el fichero de filtros standard de *Diald* es suficiente para la mayoría de los casos, aunque quizá sea poco restrictivo en determinados entornos. El `personal.filter` que se muestra a continuación tiene algunas correcciones sobre el original de la versión 0.16.

En sucesivas versiones de este documento se presentarán otros ejemplos más restrictivos comentados.

```
# /etc/diald/personal.filter
# Las reglas de filtrado a continuación son las mismas que las de
# standard.filter con las siguientes correcciones:
#
# Cambio de 10 minutos por 4 en "cualquier otra conexión TCP".
# Añadido para ignorar los FIN ACK con "ignore tcp tcp.fin".
# Ignorar paquetes icmp (con ping y traceroute no se realizará llamada).
#

# This is a pretty complicated set of filter rules.
# (These are the rules I use myself.)
#
# I've divided the rules up into four sections.
# TCP packets, UDP packets, ICMP packets and a general catch all rule
# at the end.

ignore icmp any

#-----------------------------------------------------------------------------
# Rules for TCP packets.
#-----------------------------------------------------------------------------
# General comments on the rule set:
```

```
#
# In general we would like to treat only data on a TCP link as significant
# for timeouts. Therefore, we try to ignore packets with no data.
# Since the shortest possible set of headers in a TCP/IP packet is 40 bytes,
# any packet with length 40 must have no data riding in it.
# We may miss some empty packets this way (optional routing information
# and other extras may be present in the IP header), but we should get
# most of them. Note that we don't want to filter out packets with
# tcp.live clear, since we use them later to speedup disconnects
# on some TCP links.
#
# We also want to make sure WWW packets live even if the TCP socket
# is shut down. We do this because WWW doesn't keep connections open
# once the data has been transfered, and it would be annoying to have the link
# keep bouncing up and down every time you get a document.
#
# Outside of WWW the most common use of TCP is for long lived connections,
# that once they are gone mean we no longer need the network connection.
# We don't neccessarily want to wait 10 minutes for the connection
# to go down when we don't have any telnet's or rlogin's running,
# so we want to speed up the timeout on TCP connections that have
# shutdown. We do this by catching packets that do not have the live flag set.

# --- start of rule set proper ---

# When initiating a connection we only give the link 15 seconds initially.
# The idea here is to deal with possibility that the network on the opposite
# end of the connection is unreachable. In this case you don't really
# want to give the link 10 minutes up time. With the rule below
# we only give the link 15 seconds initially. If the network is reachable
# then we will normally get a response that actually contains some
# data within 15 seconds. If this causes problems because you have a slow
# response time at some site you want to regularly access, you can either
# increase the timeout or remove this rule.
accept tcp 15 tcp.syn

# Keep named xfers from holding the link up
ignore tcp tcp.dest=tcp.domain
ignore tcp tcp.source=tcp.domain

# (Ack! SCO telnet starts by sending empty SYNs and only opens the
# connection if it gets a response. Sheesh..)
accept tcp 5 ip.tot_len=40,tcp.syn

# keep empty packets from holding the link up (other than empty SYN packets)
ignore tcp ip.tot_len=40,tcp.live

# Modificacion de Andres Seco. Ignorar los FIN ACK.
ignore tcp tcp.fin

# make sure http transfers hold the link for 2 minutes, even after they end.
# NOTE: Your /etc/services may not define the tcp service www, in which
# case you should comment out the following two lines or get a more
# up to date /etc/services file. See the FAQ for information on obtaining
# a new /etc/services file.
```

```
accept tcp 120 tcp.dest=tcp.www
accept tcp 120 tcp.source=tcp.www
# Same for https
accept tcp 120 tcp.dest=tcp.443
accept tcp 120 tcp.source=tcp.443

# Once the link is no longer live, we try to shut down the connection
# quickly. Note that if the link is already down, a state change
# will not bring it back up.
keepup tcp 5 !tcp.live
ignore tcp !tcp.live

# an ftp-data or ftp connection can be expected to show reasonably frequent
# traffic.
accept tcp 120 tcp.dest=tcp.ftp
accept tcp 120 tcp.source=tcp.ftp

#NOTE: ftp-data is not defined in the /etc/services file provided with
# the latest versions of NETKIT, so I've got this commented out here.
# If you want to define it add the following line to your /etc/services:
# ftp-data        20/tcp
# and uncomment the following two rules.
#accept tcp 120 tcp.dest=tcp.ftp-data
#accept tcp 120 tcp.source=tcp.ftp-data

# If we don't catch it above, give the link 10 minutes up time.
#accept tcp 600 any
# Modificacion de Andres Seco. Solo dejar 4 minutos mas.
accept tcp 240 any

# Rules for UDP packets
#
# We time out domain requests right away, we just want them to bring
# the link up, not keep it around for very long.
# This is because the network will usually come up on a call
# from the resolver library (unless you have all your commonly
# used addresses in /etc/hosts, in which case you will discover
# other problems.)
# Note that you should not make the timeout shorter than the time you
# might expect your DNS server to take to respond. Otherwise
# when the initial link gets established there might be a delay
# greater than this between the initial series of packets before
# any packets that keep the link up longer pass over the link.

# Don't bring the link up for rwho.
ignore udp udp.dest=udp.who
ignore udp udp.source=udp.who
# Don't bring the link up for RIP.
ignore udp udp.dest=udp.route
ignore udp udp.source=udp.route
# Don't bring the link up for NTP or timed.
ignore udp udp.dest=udp.ntp
ignore udp udp.source=udp.ntp
ignore udp udp.dest=udp.timed
ignore udp udp.source=udp.timed
```

```
# Don't bring up on domain name requests between two running nameds.
ignore udp udp.dest=udp.domain,udp.source=udp.domain
# Bring up the network whenever we make a domain request from someplace
# other than named.
accept udp 30 udp.dest=udp.domain
accept udp 30 udp.source=udp.domain
# Do the same for netbios-ns broadcasts
# NOTE: your /etc/services file may not define the netbios-ns service
# in which case you should comment out the next three lines.
ignore udp udp.source=udp.netbios-ns,udp.dest=udp.netbios-ns
accept udp 30 udp.dest=udp.netbios-ns
accept udp 30 udp.source=udp.netbios-ns
# keep routed and gated transfers from holding the link up
ignore udp tcp.dest=udp.route
ignore udp tcp.source=udp.route
# Anything else gest 2 minutes.
accept udp 120 any

# Catch any packets that we didn't catch above and give the connection
# 30 seconds of live time.
accept any 30 any
```

6.3 Realización de la llamada

Fichero `/etc/diald/diald.connect` (debe tener permiso de ejecución):

```
/usr/sbin/chat -f /etc/chatscripts/provider
```

Fichero `/etc/chatscripts/provider`. En este fichero de ejemplo que se muestra hay que revisar el numero de teléfono:

```
ABORT BUSY
ABORT "NO CARRIER"
ABORT VOICE
ABORT "NO DIALTONE"
ABORT "NO ANSWER"
"" ATZ
OK ATDT123456789
CONNECT \d\c
```

6.4 Archivo de comandos de inicio de la conexión

Debe tener permiso de ejecución.

Este archivo puede ser usado para multiples tareas (sincronizar la hora del equipo, lanzar la entrega de la cola de impresión, la recuperación de los mensajes desde la oficina de correos externa, etc.).

En el ejemplo, se envia un mensaje a `root` con los parametros que se le pasan al archivo de comandos (interface, mascara de subred, dirección IP local, dirección IP remota y prioridad en la tabla de rutas):

```
#!/bin/sh

iface=$1
```

```
netmask=$2
localip=$3
remoteip=$4
metric=$5

# Set the time and date
# netdate ntp.server.somecountry

# Run the mail queue
# runq

echo `date` $1 $2 $3 $4 $5 | mail -s "diald - conectado" root@localhost
```

7 Conexión mediante modem y PPP de un puesto aislado a varios ISPs no simultáneamente

En muchas ocasiones, un ordenador aislado no se conecta solo a una red, si no que es común acceder a diferentes redes o a Internet mediante diferentes proveedores de acceso. En estos casos, modificar los ficheros de configuración cada vez que se desea acceder a un sitio diferente puede resultar incómodo.

La solución propuesta aquí consiste en mantener diferentes juegos de ficheros de configuración para cada conexión a un proveedor diferente y se incluyen algunos archivos de comandos para automatizar el cambio de uno a otro.

7.1 Nota sobre la entrega de correo mediante un servidor de reenvío (relay host)

Tanto si su correo electrónico usa un Agente de Transferencia de Mensajes local con un servidor SMTP de reenvío (relay host) para la entrega de todos los mensajes, como si usa un cliente de correo que directamente entregue los mensajes al servidor SMTP de su proveedor de acceso, el cambio de acceso de un proveedor a otro precisará la reconfiguración de este servidor de relay, ya que los proveedores de acceso normalmente comprueban el origen de las conexiones que reciben y sólo aceptan mensajes si el buzón del destinatario está en uno de los dominios que gestiona directamente este servidor de relay o si la dirección IP del cliente que inicia la conexión pertenece a una de las direcciones que este proveedor asigna a sus clientes, para evitar que cualquiera pueda usar este servidor `smtp` para objetivos no muy limpios (spam, anónimos, enmascaramiento del origen, etc.).

En los ejemplos que se ponen a continuación se mostrara cómo hacer este cambio en los ficheros de configuración de *Smail* suponiendo una sencilla configuración en la que todos los mensajes con destino fuera del equipo local se entregan a un servidor `smtp` de reenvío (relay host). Si usted usa otro agente de transferencia de mensajes (MTA) en su sistema puede enviarme los cambios que tuvo que hacer en dicho MTA para incluirlos aquí. Igualmente si usa un cliente de correo que entrega directamente los mensajes a un servidor smtp ajeno a su sistema (Kmail, Netscape, etc.).

7.2 Archivos de comandos para automatizar la creación de las multiples conexiones y su intercambio

7.2.1 Puesta en marcha

En primer lugar, se crea un subdirectorio de `/etc/diald` llamado `providers` donde se almacenarán los archivos de comandos para automatizar el cambio y los subdirectorios con los juegos de ficheros de configuración para cada uno de los proveedores.

Con el archivo de comandos siguiente se crea este directorio y se introducen en el los ficheros de configuración de *Diald*, *chat*, *pppd* y *Smail* con los que se esté trabajando actualmente, y que serán tomados como base para las siguientes configuraciones.

```
#!/bin/sh
#Fichero /etc/diald/providers/setupdialdmultiprovider
mkdir /etc/diald/providers
mkdir /etc/diald/providers/setup
cp /etc/ppp/pap-secrets /etc/diald/providers/setup
cp /etc/ppp/chap-secrets /etc/diald/providers/setup
cp /etc/resolv.conf /etc/diald/providers/setup
cp /etc/diald/diald.options /etc/diald/providers/setup
cp /etc/diald/standard.filter /etc/diald/providers/setup
cp /etc/diald/personal.filter /etc/diald/providers/setup
cp /etc/diald/diald.connect /etc/diald/providers/setup
cp /etc/chatscripts/provider /etc/diald/providers/setup
cp /etc/diald/ip-up /etc/diald/providers/setup
cp /etc/diald/ip-down /etc/diald/providers/setup
cp /etc/smail/routers /etc/diald/providers/setup
```

7.2.2 Nuevo proveedor

Con el archivo de comandos siguiente se copia la configuración original que se estaba utilizando cuando se ejecutó `setupdialdmultiprovider` para prepararla para un nuevo proveedor o una nueva red. Este archivo de comandos (`/etc/diald/providers/newdialdprovider`) se lanza con un parámetro, el nombre del proveedor o el nombre de la red a la que accedemos.

```
#!/bin/sh
#Fichero /etc/diald/providers/newdialdprovider
mkdir /etc/diald/providers/$1
cp /etc/diald/providers/setup/* /etc/diald/providers/$1
```

Ahora será necesario modificar como corresponda los ficheros que se encuentran en el nuevo directorio `/etc/diald/providers/nombreproveedor`, siendo `nombreproveedor` el parámetro que se le ha pasado a `newdialdprovider`.

7.2.3 Cambio de uno a otro

Para finalizar, con este archivo de comandos se realizan los cambios oportunos para acceder a un proveedor o red u otro. Se utilizan enlaces simbólicos para evitar tener archivos duplicados. Así mismo, mediante enlaces simbólicos, en caso de necesitar hacer algún cambio, por ejemplo, en el fichero `/etc/resolv.conf`, modificando directamente este fichero quedará modificado el fichero al que apunta el enlace, `/etc/diald/providers/nombreproveedor/resolv.conf`.

```
#!/bin/sh
#Fichero /etc/diald/providers/setdialdprovider
/etc/init.d/diald stop
#espera para dejar tiempo a que Diald finalice.
sleep 4
ln -sf /etc/diald/providers/$1/pap-secrets /etc/ppp
ln -sf /etc/diald/providers/$1/chap-secrets /etc/ppp
ln -sf /etc/diald/providers/$1/resolv.conf /etc
ln -sf /etc/diald/providers/$1/diald.options /etc/diald
ln -sf /etc/diald/providers/$1/standard.filter /etc/diald
ln -sf /etc/diald/providers/$1/personal.filter /etc/diald
ln -sf /etc/diald/providers/$1/diald.connect /etc/diald
ln -sf /etc/diald/providers/$1/provider /etc/chatscripts
ln -sf /etc/diald/providers/$1/ip-up /etc/diald
ln -sf /etc/diald/providers/$1/ip-down /etc/diald
ln -sf /etc/diald/providers/$1/routers /etc/smail
/etc/init.d/diald start
```

8 Conexión mediante modem y PPP de un proxy/firewall a un ISP

El asunto de conectar una red a internet de modo que uno de los equipos trabaje como un servidor de conexión compartida con tareas de proxy/cache de páginas web y cortafuegos (firewall) de seguridad entre la red interna y externa es un asunto complejo que escapa al objetivo de este documento y que ya han tratado de forma fantástica otros documentos «Como» similares a este. Al final de este documento se pueden encontrar referencias y enlaces.

Aquí tan solo se va a tratar de configurar *Diald* suponiendo que el equipo ya hace IP-Masquerading, tiene un proxy como *Squid* o similar trabajando, una conexión con un ISP configurada correctamente y que la seguridad de acceso a puertos TCP/UDP ha sido revisada (fichero `/etc/inetd.conf` y algunos otros como `securetty`, `host.allow`, etc).

Básicamente, se trata de reconfigurar las reglas de enmascaramiento/filtrado/acceso cada vez que cambia el conjunto de interfaces del equipo, es decir, cuando se establece el interface ppp0 y cuando deja de existir. Un buen sitio para hacer esto son los scripts de ip-up e ip-down de *pppd*.

8.1 Ejemplo para Debian 2.1

En Debian, basta con montar el paquete *ipmasq* y durante su configuración indicar que se desea que se cambien las reglas de forma sincronizada con *pppd*. De esta manera, se crean scripts en los directorios `/etc/ppp/ip-up.d` y `/etc/ppp/ip-down.d` que llaman a `/sbin/ipmasq`, un script que analiza las interfaces actuales y hace una configuración sencilla valida en muchos casos, aunque se puede personalizar con facilidad revisando los ficheros de reglas de `/etc/ipmasq/rules`.

Tan solo es necesario realizar una corrección tras la instalación de este paquete. Se trata de cambiar el orden de ejecución del script de lanzamiento de *ipmasq* durante el arranque del equipo, eliminandolo del directorio `/etc/rcS.d` y poniendolo para que se ejecute despues de `S20diald` en `/etc/rc2.d` de modo que ya exista la interface `sl0`. `S90ipmasq` es un lugar valido para el enlace simbólico a `/etc/init.d/ipmasq`.

En Debian no es necesario preocuparse de la version del kernel, ya que el script `/sbin/ipmasq` usa `ipfwadm` o `ipchains` según corresponda.

8.2 Ejemplo para Suse 6.1

Este ejemplo se debe a Mr Cornish Rex, *troll@tnet.com.au*.

Los comandos de control de ip-masp y routing que se presentan a continuación son para kernels de la versión 2.2, mediante ipchains, pero no son válidos para kernels de las versiones 2.0. Para estos ultimos habría que traducir dichos comandos a su equivalente en el antiguo comando ipmasq.

Además, vamos a suponer que la interface ethernet del equipo tiene la dirección 192.168.1.1 con máscara de 16 bits, es decir, 255.255.0.0.

El fichero `/etc/ppp/ip-up` sería valido de la siguiente forma:

```
#!/bin/sh
# $1 = Interface
# $2 = Tty device
# $3 = speed
# $4 = local ip
# $5 = remote ip
# $6 = ipparam
/sbin/ipchains -F input
/sbin/ipchains -P input DENY
/sbin/ipchains -A input -j ACCEPT -i eth0 -s 192.168.0.0/16 -d 0.0.0.0/0
/sbin/ipchains -A input -j DENY -p udp -i $1 -s 0.0.0.0/0 -d $4/32 0:52 -l
/sbin/ipchains -A input -j DENY -p udp -i $1 -s 0.0.0.0/0 -d $4/32 54:1023 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 0:112 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 114:1023 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 6000:6010 -l
/sbin/ipchains -A input -j DENY -p icmp --icmp-type echo-request \
-i $1 -s 0.0.0.0/0 -l
/sbin/ipchains -A input -j DENY -p icmp -f -i $1 -s 0.0.0.0/0 -l
/sbin/ipchains -A input -j DENY -p udp -i $1 -s 0.0.0.0/0 -d $4/32 5555 -l
/sbin/ipchains -A input -j DENY -p udp -i $1 -s 0.0.0.0/0 -d $4/32 8000 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 8000 -l
/sbin/ipchains -A input -j DENY -p udp -i $1 -s 0.0.0.0/0 -d $4/32 6667 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 6667 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 4557 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 4559 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 4001 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 2005 -l
/sbin/ipchains -A input -j DENY -p tcp -i $1 -s 0.0.0.0/0 -d $4/32 6711 -l
/sbin/ipchains -A input -j DENY -i $1 -s 192.168.0.0/16 -d 0.0.0.0/0 -l
/sbin/ipchains -A input -j ACCEPT -i $1 -s 0.0.0.0/0 -d $4/32
/sbin/ipchains -A input -j ACCEPT -i lo -s 0.0.0.0/0 -d 0.0.0.0/0
/sbin/ipchains -A input -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 -l

/sbin/ipchains -F output
/sbin/ipchains -P output DENY
/sbin/ipchains -A output -j ACCEPT -i eth0 -s 0.0.0.0/0 -d 192.168.0.0/16
/sbin/ipchains -A output -j DENY -i $1 -s 192.168.0.0/16 -d 0.0.0.0/0 -l
/sbin/ipchains -A output -j ACCEPT -i $1 -s $4/32 -d 0.0.0.0/0
/sbin/ipchains -A output -j ACCEPT -i lo -s 0.0.0.0/0 -d 0.0.0.0/0
/sbin/ipchains -A output -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0

/sbin/ipchains -F forward
/sbin/ipchains -P forward DENY
```

```
/sbin/ipchains -M -S 120 120 120
/sbin/ipchains -A forward -j MASQ -s 192.168.1.0/24
/sbin/ipchains -A forward -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0

exit 0
```

El fichero `/etc/ppp/ip-down` sería valido de la siguiente forma:

```
#!/bin/sh
# $1 = Interface
# $2 = Tty device
# $3 = Speed
# $4 = Local ip
# $5 = Remote ip
/sbin/ipchains -F input
/sbin/ipchains -F output
/sbin/ipchains -F forward
/sbin/ipchains-restore < /etc/ppp/orig.chains
```

Y el fichero que aparece al final del script anterior, orig.chains, es el siguiente (el estado original de la configuración de ipchains):

```
# orig.chains
# creado con: ipchains-save > orig.chains
:input ACCEPT
:forward ACCEPT
:output ACCEPT
-A input -s 0.0.0.0/0.0.0.0 -d 192.168.1.1/255.255.255.255
-A output -s 192.168.1.1/255.255.255.255 -d 0.0.0.0/0.0.0.0
```

8.3 Ejemplo para Slackware 3.6

Este ejemplo se debe a Hoo Kok Mun, *hkmun@pacific.net.sg*.

Este es el ejemplo más sencillo que he visto, aunque perfectamente funcional. Desde el principio establece la norma de enmascaramiento, antes de que exista la interface `sl0`, y no cambia cuando se establece `ppp0`. Si son necesarias medidas de seguridad, probablemente quede un poco limitado.

```
#/etc/rc.d/rc.local
/sbin/ipfwadm -F -p deny
/sbin/ipfwadm  -F -a m -S 192.168.0.0/24 -D 0.0.0.0/0
```

Como se puede ver, es para kernels de la versión 2.0.

9 Programas y versiones utilizados

Para el desarrollo de esta documentación he utilizado las siguientes versiones de diald:

- Diald 0.16.5 - La ultima versión que mantuvo el autor original de diald.
- Diald 0.99.3 - La ultima versión aparecida hasta el momento de la primera redacción de este documento.

Y las siguientes versiones de pppd:

- pppd 2.3.5

La versión 0.16.5 de Diald es seguramente la más extendida, y la que se incluye directamente con muchas de las distribuciones de linux. Es suficiente en la mayoría de los casos y resulta muy estable, aunque, sin duda, las más modernas versiones, aunque se encuentran en desarrollo, incluyen funcionalidades muy interesantes.

10 Más información

La documentación original a partir de la cual ha sido obtenido este documento puede ser encontrada en las paginas man de `diald`, `diald-examples`, `diald-control`, `diald-monitor`, `dctrl`, `pppd`, `chat`, en los directorios correspondientes de /usr/doc y en las paginas de los paquetes en la World wide web:

- Nueva Página Oficial de Diald: *http://diald.sourceforge.net/*
- Descarga de nuevas versiones: *ftp://diald.sourceforge.net/pub/diald/*
- Página anterior de Diald: *http://diald.unix.ch*
- Página antigua de Diald hasta la versión 0.16.5: *http://www.loonie.net/~erics/diald.html*
- Antiguo sitio FTP para nuevas versiones de pppd: *ftp://cs.anu.edu.au/pub/software/ppp/*

Existe una lista de correo para hablar y discutir sobre diald en el servidor de listas de David S. Miller. Para suscribirse, enviar un mensaje con la linea subscribe linux-diald» EN EL CUERPO DEL MENSAJE a la dirección *Majordomo@vger.rutgers.edu*.

Existen también múltiples documentos RFC (Request For Comments) que detallan cómo debe ser el funcionamiento de las lineas con encapsulado PPP y sus protocolos asociados (LCP, IPCP, PAP, CHAP, ...), y pueden ser encontradas en /usr/doc/doc-rfc y en diversos lugares de la World wide web, como *http://metalab.unc.edu* y *http://nic.mil/RFC*. Puedes solicitar información sobre las RFCs en *RFC-INFO@ISI.EDU*.

Y también pueden resultar de ayuda los «Comos» siguientes:

- DNS-HOWTO - *http://www.linuxdoc.org/HOWTO/DNS-HOWTO.html* (traducción: *http://www.insflug.org/documentos/DNS-Como/*)
- Firewall-HOWTO - *http://www.linuxdoc.org/HOWTO/Firewall-HOWTO.html* (traducción: *http://www.insflug.org/documentos/Cortafuegos-Como/*)
- IP-Masquerade-HOWTO - *http://www.linuxdoc.org/HOWTO/IP-Masquerade-HOWTO.html* (traducción: *http://www.insflug.org/documentos/IP-Masquerade-Como/*)
- IPCHAINS-HOWTO - *http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html*
- Modem-HOWTO - *http://www.linuxdoc.org/HOWTO/Modem-HOWTO.html*
- NET3-4-HOWTO - *http://www.linuxdoc.org/HOWTO/NET3-4-HOWTO.html* (traducción: *http://www.insflug.org/documentos/Redes-En-Linux-Como/*)
- PPP-HOWTO - *http://www.linuxdoc.org/HOWTO/PPP-HOWTO.html* (traducción: *http://www.insflug.org/documentos/PPP-Como/*)
- Serial-HOWTO - *http://www.linuxdoc.org/HOWTO/Serial-HOWTO.html* (traducción: *http://www.insflug.org/documentos/Serie-Como/*)

11 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos, así como de la producción de documentos originales en aquellos casos en los que no existe análogo en inglés, centrándose, preferentemente, en documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. :)), etc.

Diríjase a la sede del Insflug para más información al respecto.

En ella encontrará siempre las **últimas** versiones de las traducciones «oficiales»: *www.insflug.org*. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Además, cuenta con un sistema interactivo de gestión de fe de erratas y sugerencias en línea, motor de búsqueda específico, y más servicios en los que estamos trabajando incesantemente.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

En *http://www.insflug.org/insflug/creditos.php3* cuenta con una detallada relación de las personas que hacen posible tanto esto como las traducciones.

¡Diríjase a *http://www.insflug.org/colaboracion/index.php3* si desea unirse a nosotros!.

«Cartel» Insflug, *cartel@insflug.org*.