

## El día a día del Administrador de Sistemas: Portfwd

# DIRECCIONES NUEVAS

Las conexiones entrantes TCP no siempre terminan donde debieran. Un redireccionador configurable gratuito arroja la basura digital hacia otra dirección.

**POR CHARLY KÜHNAST**

**H**ace poco hablábamos sobre Rinetd, un redireccionador de TCP. Rinetd es poco exigente, de confianza y fácil de configurar, pero carece de características avanzadas. Este mes examinaré Portfwd, una herramienta que incluye algunos de aquellos rasgos que fallan en Rinetd. Portfwd (Port Forwarding Daemon, [1]) tiene forma de fichero tar de 116K, que se puede compilar e instalar de la forma normal:

```
./configure; make; make install
```

o parecido. El binario, de fundamental importancia, llegó a ocultarse en el directorio *src* tras teclear *make install*. Un enlace simbólico (*ln -s /usr/local/portfwd-0.27 /src/portfwd/usr/bin/portfwd*) se encargó de esto.

## Sencillo

Primero vamos a comprobar cómo Portfwd maneja el cambio de dirección simple. Queremos reenviar cualquier paquete TCP que llega por el puerto 80 al servidor en 10.20.30.40. El fichero de configuración se parece a esto:

```
user nobody
group nobody
tcp { 80 { => 10.20.30.40:80 } }
```

Las llaves externas contienen el puerto donde Portfwd recibe la conexión entrante y las llaves internas se encargan de redirigirla. Por supuesto, los mejores Rinetd también podrían haber manejado esto. Pero, me dí cuenta de que era capaz de configurar múltiples tarjetas para que utilizaran Portfwd como un simple balanceador de carga round-robin. Cambié el primer ejemplo para reenviar las conexiones entrantes del puerto 80 a dos servidores, 10.20.30.40 y 10.20.30.41, de la siguiente forma:

```
tcp { 80 { =>
10.20.30.40:80,
10.20.30.41:80 } }
```

Si mi equipo está conectado a dos o más redes o posee dos o más direcciones de red (multihomed), puedo utilizar *bind-address* para especificar la interfaz. Para mi tercer experimento, le asigné las direcciones 192.168.1.1 y 192.168.1.2 a mi equipo. Quería presentar cualquier conexión entrante para el puerto 25 a los otros dos servidores:

```
bind-address 192.168.1.1
tcp { 25 { => 10.20.30.40:25 } }
bind-address 192.168.1.2
tcp { 25 { => 10.20.30.41:25 } }
```

A diferencia de Rinetd, Portfwd no sólo reenvía conexiones TCP sino también



datagramas UDP. La siguiente regla le indica a Portfwd que acepte peticiones DNS y reenvíe estas peticiones al servidor *zdns.example.com*:

```
udp { 53 { => z
dns.example.com:53 } }
```

## Frágil

Portfwd también me proporciona la posibilidad de definir un caso especial para servidores con conexiones que tienden a romperse con bastante frecuencia. La clave *fragile* le dice a Portfwd que sea compasivo en el caso de errores de conexiones – y que lo vuelva a intentar con más frecuencia de lo habitual:

```
fragile tcp { 80 { => z
dialup.example.com:8080 } }
```

Este es el tipo de configuración que me gusta utilizar cuando quiero que una máquina central parezca un monstruo de carga de funciones del exterior, aunque en realidad la mayor parte de esta funcionalidad se descargue en máquinas separadas según como dicten las necesidades. ■

## SYSADMIN

X Window..... 68

El sistema X Window.

TCP Hijacking..... 71

Previendo ataques TCP.

## RECURSOS

[1] Portfwd: <http://portfwd.sf.net>