

El Día a Día del Administrador de Sistemas: **ssld**

LOS SERVICIOS EN DETALLE

Algunos de los servidores de Charly ejecutan el servicio SSH en el puerto 443 en vez de en el puerto estándar 22. Si un servidor web Apache con capacidad SSL arranca con problemas, la solución se llama `sslh`.

POR CHARLY KÜHNAST

```

root@funghi: /home/charly
root@salami: lsof | grep TCP

[...]
sshd      1345      root      3u      IPv4      5676      0t0      TCP localhost:ssh (LISTEN)
apache2   2520     www-data  4u      IPv6      7490      0t0      TCP localhost:https (LISTEN)
sslh      3328      sslh      3u      IPv4      19957     0t0      TCP 10.50.5.42:https (LISTEN)
[...]

charly@funghi:~$ ssh -p 443 charly@10.50.5.42
charly@10.50.5.42's password:
Linux funghi 2.6.31-14-generic #48-Ubuntu SMP Fri Oct 16 14:04:26 UTC 2009 i686
charly@funghi:~$

```

Figura 1: En lugar de que Apache y SSH se peleen por el puerto 443, el servicio `sslh` identifica el tipo de solicitud (SSH en este caso) y la pasa al servicio responsable de la misma.

Tanto si estoy en un cibercafé, como utilizando la red inalámbrica en un hotel o utilizando un punto de acceso público en un aeropuerto, continuamente me encuentro bloqueado tras un cortafuegos que rechaza conexiones al puerto 22. Por supuesto, ningún cortafuegos bloqueará el tráfico a los puertos 80 y 443.

En otras palabras, es una buena idea enlazar el servicio SSH al puerto HTTPS en mis servidores. Esto me ahorra tener que hacer un túnel y puedo simplemente registrarme en mis servidores con `ssh -p 443 <user> @ <host>`. Pero si el puerto del HTTPS está ocupado por un servidor web con SSL habilitado, tengo que ponerme a pensar qué hacer. Use `sslh` [1].

Los creadores de esta herramienta la llamaron multiplexor SSL/SSH. La idea subyacente es que el multiplexor escuche en el puerto 443 y descubra las conexiones entrantes si el cliente quiere hablar HTTPS con SSH al host. Los servicios por sí mismos están ligados al `localhost:443` y `localhost:22`, respectivamente (Figura 1). `sslh` obtiene esta información del fichero `/etc/defaults/sslh`, que se parece a la siguiente configuración

```

RUN=yes
DAEMON_OPTS="-u sslh \
-p 10.50.5.42:443 \
-s 127.0.0.1:22 \
-l 127.0.0.1:443 \
-P /var/run/sslh.pid"

```

Para saber qué protocolo se requiere en cada momento, `sslh` analiza el comportamiento del cliente. En el caso de una conexión HTTPS entrante, el cliente espera a que el servidor envíe la señal que está lista para recibir. Una respuesta de un cliente en una conexión SSH no esperará, pero abrirá el diálogo por sí misma, y `sslh` esperará un

corto tiempo, normalmente dos segundos. Si el cliente no envía ningún dato en este tiempo, `sslh` supondrá que es una conexión HTTPS y la reenviará al servidor web con `127.0.0.1:443`.

dominio de Apache

Para restringir el servidor Apache al `localhost`, tengo que cambiar el parámetro de la lista en su configuración que establece SSL al 443 por defecto a `127.0.0.1:443`. Estrictamente hablando, no tiene que ligar el puerto SSH al `localhost` necesariamente, ya que no hay conflicto con otro servicio en el puerto 22. Sin embargo, lo hice así por dos motivos: Primero, me protege a mí y a mi `auth.log` de una gran cantidad de comprobaciones que continúan apareciendo en esa dirección. Segundo, puedo llegar al servidor mediante la consola serie si SSH o `sslh` fallasen por algún motivo.

RECURSOS

[1] `sslh`: <http://www.rutschle.net/tech/sslh>

SYSADMIN

GRUB262

Una nueva imagen y varias características nuevas para el ubicuo gestor de arranque GRUB.

BleachBit65

Bleachbit limpia los archivos de registro innecesarios conveniente y eficazmente.

EL AUTOR

Charly Kühnast es Gerente de Sistemas Unix en el centro de datos de Moers, Alemania, cerca del conocido Rin. Entre sus labores se



incluye la seguridad del cortafuegos, la disponibilidad y el cuidado de la DMZ (zona desmilitarizada). Divide su tiempo libre entre el calor, la humedad y oriente, donde se divierte cocinando, visitando acuarios y aprendiendo japonés respectivamente.