Servicios de red xinetd

1. Presentación

El demonio **xinetd** es un "superservicio" que permite controlar el acceso a un conjunto de servicios, **telnet** por ejemplo. Se pueden configurar muchos servicios de red para funcionar con **xinetd**, como los servicios FTP, ssh, samba, rcp, HTTP, etc. Se pueden aplicar opciones de configuración específicas para cada servicio gestionado.

La llegada de systemd, que permite controlar a la vez un socket y un servicio, hace obsoleto el uso de xinetd en las distribuciones recientes. Sin embargo, xinetd sigue estando presente en todas las distribuciones. Esta parte ha sido escrita usando una antigua versión de OpenSUSE, ya que xinetd ya no está instalado por defecto.

Cuando un cliente se conecta a un servicio de red controlado por **xinetd**, xinetd recibe la petición y verifica primero las autorizaciones de acceso TCP (vea **tcp_wrappers** en el próximo capítulo); luego, las reglas definidas para este servicio (autorizaciones específicas, recursos asignados, etc.). El demonio levanta una instancia del servicio y le cede la conexión. A partir de entonces, **xinetd** ya no interfiere en la conexión entre el cliente y el servidor.

2. Configuración

Los archivos de configuración son:

- **/etc/xinetd.conf**: configuración global
- **/etc/xinetd.d/***: directorio que contiene los archivos específicos para los servicios. Existe un archivo por servicio, con el mismo nombre que el especificado en `/etc/services` .

```
$ ls -l /etc/xinetd.d
total 92
-rw-r--r-- 1 root root  313 sep 22  2007 chargen
```

```
-rw-r--r-- 1 root root  333 sep 22  2007 chargen-udp
-rw-r--r-- 1 root root  256 mar 20 22:11 cups-lpd
-rw-r--r-- 1 root root  409 nov  4  2005 cvs
-rw-r--r-- 1 root root  313 sep 22  2007 daytime
-rw-r--r-- 1 root root  333 sep 22  2007 daytime-udp
-rw-r--r-- 1 root root  313 sep 22  2007 discard
-rw-r--r-- 1 root root  332 sep 22  2007 discard-udp
-rw-r--r-- 1 root root  305 sep 22  2007 echo
-rw-r--r-- 1 root root  324 sep 22  2007 echo-udp
-rw-r--r-- 1 root root  492 sep 22  2007 netstat
-rw-r--r-- 1 root root  207 abr 23 19:04 rsync
-rw-r--r-- 1 root root  337 feb 17 14:22 sane-port
-rw-r--r-- 1 root root  332 sep 22  2007 servers
-rw-r--r-- 1 root root  334 sep 22  2007 services
-rw-r--r-- 1 root root  351 jun 21  2007 svnserve
-rw-r--r-- 1 root root  277 nov  8  2007 swat
-rw-r--r-- 1 root root  536 sep 21  2007 systat
-rw-r--r-- 1 root root  387 feb  4 10:11 tftp.rpmsave
-rw-r--r-- 1 root root  339 sep 22  2007 time
-rw-r--r-- 1 root root  333 sep 22  2007 time-udp
-rw-r--r-- 1 root root 2304 abr  4 11:39 vnc
-rw------- 1 root root  768 sep 22  2007 vsftpd
```

Contenido de xinetd.conf:

```
defaults
{
    instances            = 60
    log_type             = SYSLOG authpriv
    log_on_success       = HOST PID
    log_on_failure       = HOST
    cps                  = 25 30
}
includedir /etc/xinetd.d
```

- **instances**: número máximo de peticiones que un servicio **xinetd** puede gestionar en un instante dado.

- **log_type**: en nuestro caso, el demonio **syslog** gestiona las trazas mediante **authpriv** y las trazas están colocadas en /var/log/secure. **FILE /var/log/xinetd** hubiera colocado las trazas en /var/log/xinetd.
- **log_on_success**: **xinetd** va a registrar el evento si la conexión al servicio tiene éxito. La información trazada son el cliente (**HOST**) y el **ID** del proceso servidor que trata de la conexión.
- **log_on_failure**: igual para los fracasos. Resulta fácil saber qué clientes han intentado conectarse si, por ejemplo, no se autoriza la conexión.
- **cps**: **xinetd** sólo autoriza 25 conexiones por segundo a un servicio. Si se alcanza el límite, **xinetd** esperará 30 segundos antes de autorizar de nuevo las conexiones.
- **includedir**: incluye las opciones de los archivos presentes en el directorio indicado.

Ejemplo /etc/xinetd.d/telnet:

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
      disable        = no
      flags          = REUSE
      socket_type    = stream
      wait           = no
      user           = root
      server         = /usr/sbin/in.telnetd
      log_on_failure  += USERID
}
```

La primera línea de comentario, **default**, tiene una importancia particular. No la interpreta **xinetd**, sino **ntsysv** o **chkconfig**, para determinar si el servicio está activo.

- **service**: nombre del servicio que corresponde a un servicio definido en `/etc/services` .
- **flags**: atributos para la conexión. REUSE indica que se volverá a utilizar el socket

para una conexión telnet.

- **socket_type**: especifica el tipo de socket. En general, **stream** (tcp) o **dgram** (udp). Una conexión directa IP se hace por **raw**.
- **wait**: indica si el servidor es single-threaded (yes) o multi-threaded (no).
- **user**: con qué cuenta de usuario se iniciará el servicio.
- **server**: ruta del ejecutable que se debe iniciar.
- **log_on_failure**: el **+=** indica que se añade la opción asociada al archivo de traza, además de las opciones por defecto. Aquí: el login.
- **disable**: indica si el servicio está activo o no.

Algunas opciones pueden mejorar las condiciones de acceso y la seguridad:

- **only_from**: permite el acceso únicamente a los anfitriones especificados.
- **no_access**: impide el acceso a los anfitriones especificados (p. ej.: 172.16.17.0/24).
- **access_times**: autoriza el acceso únicamente en una franja horaria dada (p.ej.: 09:00-18:30).

3. Inicio y parada de los servicios

Se distinguen dos casos: systemctl y los antiguos métodos.

Primer caso, utilice, si xinetd está instalado en su servidor, el comando **systemctl** habitual:

```
# systemctl start xinetd
```

El estado se obtiene con:

```
# systemctl status xinetd
...
ene. 31 22:05:05 client xinetd[3786]: Started working: 0 available services
```

xinetd se arranca y no genera ahora ningún servicio. Agregar o eliminar servicios se efectúa editando los archivos **/etc/xinetd.d/***, modificando el parámetro **enable** (yes, no) y recargando el servicio xinetd.

El **segundo caso**, el servicio **xinetd** es un servicio como otro cualquiera cuyo inicio o cuya parada puede efectuarse con el comando **service** o directamente mediante la ejecución de **/etc/init.d/xinetd**.

```
# service xinetd start
```

En este caso, el comando **chkconfig** (Red Hat, OpenSUSE) autoriza o no el arranque del servicio al inicio para cada nivel de ejecución (runlevel).

```
# chkconfig --level 345 xinetd on
```

Como **xinetd** gestiona varios servicios, la parada de **xinetd** detiene todos los servicios asociados y el lanzamiento de **xinetd** inicia todos los servicios asociados. No es posible elegir qué servicios de xinetd se han iniciado en tal o cual nivel de ejecución o para un determinado objetivo de systemd.

```
# chkconfig telnet on
```

4. Conversión hacia systemd

Cualquier servicio xinetd se puede arrancar directamente gracias a systemd. El único problema es que el socket (la apertura del puerto) la hace xinetd y no el servicio. Es necesario que systemd cree un socket antes de lanzar el servicio. He aquí un ejemplo con tftpd, protocolo simplificado (trivial) de tranmisión de archivos.

Tftp funciona en UDP en el puerto 69. Cree el archivo **`/etc/systemd/system/tftp.socket`** , con este contenido, `ListenDatagram` solicitando la creación de un socket UDP en el puerto 69:

```
[Unit]
Description=Tftp Socket

[Socket]
ListenDatagram=69

[Install]
WantedBy=sockets.target
```

Después, cree el archivo **`/etc/systemd/system/tftp.service`** :

```
[Unit]
Description=Tftp Server
Requires=tftp.socket

[Service]
ExecStart=/usr/sbin/in.tftpd -s /var/lib/tftpboot
StandardInput=socket

[Install]
Also=tftp.socket
```

Arranque el servicio tftp, el cual abrirá automáticamente el socket y se pondrá en espera de la conexión:

```
# systemctl start tftp
# systemctl status tftp
• tftp.service - Tftp Server
   Loaded: loaded (/usr/lib/systemd/system/tftp.service; indirect; vendor
preset: disabled)
   Active: active (running) since Sun 2020-02-09 20:34:52 CET; 2s ago
# systemctl status tftp.socket
• tftp.socket - Tftp Server Activation Socket
   Loaded: loaded (/usr/lib/systemd/system/tftp.socket; disabled; vendor
preset: disabled)
   Active: active (running) since Sun 2020-02-09 20:34:52 CET; 5s ago
   Listen: [::]:69 (Datagram)
```