Control de acceso a los servicios I: Xinetd y TCPWrappers.

Autor: Enrique V. Bonet Esteban

Introducción.

Las versiones actuales de Linux[1] controlan el acceso a los servicios que ofrecen mediante los envolventes de acceso, conocidos comúnmente como TCPWrappers, y mediante el servidor de servidores xinetd. Ambos servicios no son más que unas versiones mejoradas de los antiguos envolventes de acceso y del antiguo servidor inetd que poseían los sistemas UNIX.

Por tanto, dado que los métodos de control de acceso a los servicios son una evolución de sistemas anteriores, describiremos previamente estos sistemas, para pasar con posterioridad a describir los nuevos sistemas de control de acceso existentes.

El servidor inetd.

Hasta la versión UNIX 4.2 de Berkeley Software Design (BSD), los servicios de red se encontraban ejecutándose en todo momento, escuchando en los puertos correspondientes y consumiendo con ello muchos recursos. En la versión 4.3 de BSD[2] esto se modificó, introduciendo un nuevo servidor responsable de gestionar otros servicios de la red.

Este nuevo servidor, conocido como *inetd*, gestiona todos los demonios de los servicios de red de acuerdo con las instrucciones que se le proporcionan en su archivo de configuración, generalmente */etc/inetd.conf*, mejorando las prestaciones y simplificando el proceso de arranque de otros servicios.

El archivo */etc/inetd.conf* es leído por el demonio *inetd* cuando se arranca por primera vez, generalmente en el arranque del sistema, o bien como respuesta a la señal de *hangup*[3]. El servidor *inetd* escucha las posibles conexiones a los puertos bien conocidos y arranca los demonios apropiados, entre los cuales suele estar incluido el servicio de llamada a procedimiento remoto (Remote Procedure Call), el servicio de telnet, etc.

En caso de que la solicitud de conexión sea de una aplicación orientada a conexión (protocolo de transporte TCP), el demonio *inetd* crea un nuevo proceso para gestionar la comunicación. El nuevo proceso ejecuta la aplicación orientada a conexión que atiende dicho puerto, según se especifica en el archivo */etc/inetd.conf*, le comunica la dirección IP y puerto del ordenador que ha solicitado el servicio y a continuación deja que la aplicación ejecutada atienda la petición.

Si la solicitud de conexión corresponde a un servicio no orientado a conexión (protocolo de transporte UDP), *inetd* crea un nuevo proceso cuando recibe el datagrama.

[1] Recordar que todos los ejemplos, comentarios, etc., de los apuntes se refieren a la distribución Fedora 17.
[2] La versión UNIX 4.3 de BSD apareció en el año 1985.
[3] Generalmente corresponde a la señal de valor 1 y nombre simbólico SIGHUP.

Este proceso libera el socket original para que *inetd* pueda recibir posteriores datagramas, y utiliza el socket existente para procesar tanto el datagrama recibido como nuevos datagramas que lleguen por ese socket, si estos son necesarios. Mientras el programa procesa el datagrama el servicio de *inetd* permanece en espera.

Un ejemplo de fichero de configuración */etc/inetd.conf* es el siguiente[4]:

```
ftp       stream   tcp   nowait   root   /usr/etc/ftpd      ftpd –l
telnet    stream   tcp   nowait   root   /usr/etc/telnetd   telnetd
daytime   dgram    udp   wait     root   internal
```

En el fichero pueden verse siete campos. El primero indica el nombre del servicio[5], el segundo indica el tipo de socket (stream, dgram ó raw), mientras que el tercero indica el protocolo de transporte que utiliza[6] (TCP, UDP, etc.). El cuarto campo indica si el servicio de *inetd* debe o no esperar a que el proceso que atiende la petición termine o no su ejecución antes de continuar, pudiendo observarse, como indicamos con anterioridad, que para los servicios TCP el servidor de *inetd* no espera a que termine la ejecución (valor *nowait*), mientras que para los servicios UDP el servidor *inetd* espera la terminación (valor *wait*). El quinto campo especifica bajo que usuario se ejecutará el servicio, mientras que el sexto indica la localización del programa que ejecuta ese servicio[7] y el séptimo campo son los argumentos que han de pasarse en la llamada al servicio, comenzando por el propio nombre del programa.

Los envolventes de acceso.

Los envolventes de acceso son programas que permiten controlar y limitar, si se desea, el acceso a los servicios que, con protocolo de transporte TCP ó UDP, son proporcionados por un ordenador. Aunque dicho control podría ser realizado servicio a servicio, los envolventes de acceso permiten controlar, mediante un solo programa, el acceso a los servicios TCP ó UDP, lo cual proporciona ventajas evidentes como:

- Solo existe un programa que controle el acceso, por lo que cualquier error de programación, etc., detectado en el mismo, debe ser modificado solo en él y no en todos los programas, facilitando el mantenimiento del sistema y de su seguridad.

- Todas las condiciones de acceso se encuentran en unos pocos ficheros, siendo por tanto mucho más sencillo mantenerlos y comprobar la consistencia lógica de las condiciones incluidas en los mismos.

- La información de los accesos al ordenador es escrita en un conjunto reducido de ficheros, con lo cual su auditoria es mucho más sencilla de realizar.

[4] Este fichero y los posteriores comentarios corresponden a un sistema operativo UNIX-IRIX de Silicon Graphics. Además, el fichero original contiene muchas más entradas de formato similar a las aquí mostradas.
[5] La relación entre el nombre del servicio y el puerto que atiende se encuentra en el fichero */etc/services*.
[6] El protocolo debe ser uno de los especificados en el fichero */etc/protocols*.
[7] El valor *internal* en este campo especifica que el servicio es proporcionado directamente por el servidor de inetd.

Existen multitud de envolventes de acceso, unos para el correo electrónico, otros para los servicios proporcionados por inetd, etc., pero el envolvente de acceso más conocido, y que ha proporcionado el nombre por el que son conocidos este tipo de programas, es TCPWrapper.

TCPWrapper es un envolvente de acceso para el servidor inetd que puede obtenerse, mediante el protocolo ftp y usuario anónimo, en la dirección de Internet *ftp.porcupine.org/pub/security/tcp_wrappers_X.Y.tar.gz*, donde X e Y indican la versión del programa TCPWrapper.

Los envolventes de acceso en general, y TCPWrapper en particular, permiten ejecutar las siguientes acciones:

- Mostrar mensajes de acceso (banners) al cliente que intenta acceder a un servicio.
- Realizar una búsqueda inversa doble[8] de la dirección IP del cliente, cortando la conexión en caso de discrepancia en la información obtenida.
- Registrar información de los accesos autorizados y denegados.
- Controlar el acceso al servicio solicitado en función del cliente, transfiriendo el control al verdadero servidor de red en caso de ser aceptado el acceso, o bien, en caso de ser denegado, rechazar el acceso de forma explícita, generalmente cortando la conexión, o bien transferir el control a un "falso servidor"[9].

La instalación de los envolventes de acceso suele ser tan sencilla como sustituir en el fichero de configuración de *inetd*, esto es, en */etc/inetd.conf*, el nombre del servidor por el del envolvente de acceso, generalmente de nombre *tcpd*, colocando como parámetro el nombre del verdadero servidor de red. Así, por ejemplo la línea del servicio de telnet, vista en el ejemplo anterior del fichero */etc/inetd.conf*, quedaría modificada como:

```
telnet    stream    tcp    nowait    root    /usr/etc/tcpd    telnetd
```

Las reglas de configuración de los permisos de acceso al sistema se encuentran, generalmente, en dos ficheros, */etc/hosts.allow* y */etc/hosts.deny*. El fichero */etc/hosts.allow* contiene las reglas de los servicios que son permitidos y a que ordenadores se permite cada servicio, mientras que el fichero */etc/hosts.deny* contiene las reglas de los servicios que son denegados y a que ordenadores son denegados. El orden en que se examinan dichos ficheros son primero */etc/hosts.allow* y luego */etc/hosts.deny*. Si un servicio para un ordenador se encuentra permitido en */etc/hosts.allow* se permite el uso del servicio a ese ordenador, si no es así se examina el fichero */etc/hosts.deny* en busca de si dicho permiso esta denegado explícitamente para ese ordenador, en caso de no encontrase denegado explícitamente y, ante la falta de

[8] Una búsqueda inversa doble consiste en obtener, a partir de la dirección IP del cliente su nombre y, a partir de ese nombre, obtener otra vez la dirección IP del cliente, comparando la dirección IP inicial y la obtenida para comprobar su veracidad.

[9] Esta opción permite observar el comportamiento del intruso y determinar su estrategia de intrusión, herramientas que utiliza, propósito de la misma, etc.

reglas de permiso o denegación, el acceso es permitido[10]. Esta política de permiso por defecto debe ser tenida en cuenta a la hora de establecer la seguridad del sistema.

El servidor xinetd.

Como comentamos en la introducción del tema, el servidor *xinetd* es el servidor que se encarga, en las versiones actuales de Linux, de ofrecer de forma conjunta los servicios proporcionados por el servidor *inetd* y los envolventes de acceso para todos aquellos servicios que son ejecutados por el mismo.

El servidor *xinetd* se encuentra en el directorio */usr/sbin* y, en la configuración habitual de los ordenadores, suele arrancarse en el inicio del sistema, aunque es posible arrancarlo mediante el comando:

```
systemctl start xinetd.service
```

En el nuevo servidor *xinetd* la configuración ha sufrido un notable incremento de complejidad respecto a la del primitivo servidor *inetd*, pasando de ser un sencillo archivo en el formato descrito para *inetd*, a un fichero con unos parámetros de configuración por defecto y un directorio que contiene ficheros con parámetros de configuración particulares para cada uno de los demonios de red que gestiona *xinetd*.

El fichero de configuración por defecto de *xinetd* es el fichero */etc/xinetd.conf*. Dicho fichero contiene la configuración común para todos los servidores que gestiona el demonio *xinetd*. Las entradas del fichero son de la forma:

```
service <nombre del servicio>
{
      <atributo> <operador> <valor>...
      ...
}
```

Donde <*nombre del servicio*> identifica el nombre del servicio que configura esta entrada (ftp, telnet, etc.), <*atributo*> especifica el atributo que se esta configurando, <*operador*> puede ser ‘=’, ‘+=’ y ‘-=’, que asignan, añaden y eliminan valores del atributo respectivamente[11], y <*valor*> es el valor dado al atributo. Los atributos, así como los valores que pueden tomar y una breve descripción de los mismos se encuentran a continuación, habiendo sido clasificados según la función de cada atributo.

Opciones generales:

Atributo	Descripción
id	Identifica de forma univoca el servicio. Este atributo tiene por defecto el nombre del servicio y, de forma general, solo es necesario cuando un mismo servicio posee diferentes protocolos y necesita ser descrito con diferentes entradas en el fichero de configuración.

[10] Existen palabras, etc., que permiten especificar de forma simultánea muchos servicios, direcciones IP, etc. Todo esto será descrito cuando expliquemos la configuración de los envolventes de acceso en Linux.
[11] La mayoría de atributos solo admiten el operador ‘=’.

Atributo	Descripción
type	Identifica el tipo de servicio y puede ser una combinación de los valores RPC (servicio RPC), INTERNAL (servicio proporcionado por el propio *xinetd*), TCPMUX/TCPMUXPLUS (servicio que debe ser arrancado de acuerdo al protocolo descrito en el RFC 1078 en un puerto TCPMUX bien conocido) y UNLISTED (servicio no listado en los ficheros estándar de servicios del ordenador).
flags	Identifica el modo de funcionamiento del servicio y es una combinación de los valores INTERCEPT (interceptar los paquetes o aceptar conexiones para verificar que provienen de ordenadores validos), NORETRY (no reintentar en caso de que falle la llamada a la creación de un proceso hijo mediante *fork*), IDONLY (aceptar conexiones solo cuando el ordenador remoto identifique al usuario remoto), NAMEINARGS (colocar el nombre del servicio como argumento primero en la llamada al servidor, tal y como sucede con *inetd*), NODELAY (permite que si el servicio es de tipo TCP, pueda configurarse la opción TCP_NODELAY en el socket), KEEPALIVE (permite que si el servicio es de tipo TCP, pueda configurarse la opción SO_KEEPALIVE en el socket), NOLIBWRAP (desactiva la llamada interna al TCPWrapper, con lo cual la llamada debe ser realizada de forma explícita como sucedía con *inetd*), SENSOR (reemplaza el servicio con un sensor que detecta los accesos al puerto especificado), IPv4 (especifica que el servicio es de tipo IPv4, esto es, AF_INET) y por último IPv6 (especifica que el servicio es de tipo IPv6, esto es, AF_INET6).
include	Indica el nombre de un fichero que será tomado como nuevo fichero de configuración.
includedir	Indica el nombre de un directorio cuyos ficheros serán añadidos como configuración de *xinetd*. De estos ficheros se excluye todos aquellos que contienen un punto en su nombre o terminan con una tilde (~).
disable	Es un valor booleano (“yes” o “no”) que indica si el servicio esta habilitado o deshabilitado.
enabled	Toma como argumento la lista de los *id* que deben ser habilitados. Aquellos que no se encuentren en esta lista serán deshabilitados.
socket_type	Especifica el tipo de socket, sus valores son *stream*, *dgram*, *raw* y *seqpacket* (secuencia de datagramas).
protocol	Especifica el protocolo que emplea el servicio. Sus valores posibles son cualquier protocolo de transporte especificado en */etc/protocols*.
wait	Sus valores posibles son “yes” o “no” e indica si *xinetd* debe esperar la finalización del servidor de ese servicio antes de lanzar otro servidor (valor “yes”) o no (valor “no”).
user	Determina el UID con el que se ejecutara el proceso. Dicho UID debe existir en el fichero */etc/password*.
group	Determina el GID del proceso servidor. Si el GID no existe se utiliza el GID del usuario.
server	Indica el nombre del programa que ejecuta este servicio.
server_args	Determina los argumentos que se pasaran al servidor.
rpc_versión	Determina la versión de RPC para un servicio RPC. La versión puede ser un número o un rango en el formato número-número.
rpc_number	Determina el número para un UNLISTED RPC.
env	Indica una lista de strings en formato ‘nombre=valor’. Esos strings serán añadidos a las variables de ambiente antes de arrancar el servidor.
passenv	Determina la lista de las variables de ambiente de *xinetd* que deben ser pasadas al proceso servidor.
port	Determina el puerto del servicio[12].

[12] Si este atributo es especificado para un servicio listado en el fichero /etc/services, debe ser igual al valor indicado en ese fichero.

Atributo	Descripción
redirect	Permite a los servicios TCP ser redirigidos a otro ordenador. Cuando *xinetd* recibe una conexión TCP a ese puerto, establece una conexión con el ordenador y puerto especificado y envía todos los datos entre los dos ordenadores. La sintaxis es *redirect = <dirección IP> <puerto>*[13].
bind	Permite al servicio ser asignado a una determinada dirección IP o interface específico del ordenador, esto permite, por ejemplo, que el servicio este disponible para un interface de la intranet y no para el interface que da acceso a Internet, su sintaxis es *bind = <dirección IP o interface>*.
interface	Es un sinónimo de *bind*.
groups	Puede tomar los valores “yes” o “no” e indica si el servidor es ejecutado con los permisos de los grupos a los que el UID del servidor tiene acceso o no.
umask	Especifica la máscara del servicio en formato octal. La máscara por defecto es 022.

Opciones de limitación de acceso:

Atributo	Descripción
only_from	Indica que ordenadores están autorizados a ejecutar este servicio en particular. Las formas más comunes de especificación son mediante una dirección IP concreta (147.156.222.65), un rango de direcciones IP especificado en formato dirección/rango de la mascara (147.156.222.0/23), el nombre de un ordenador (*glup.irobot.uv.es*) o el nombre de un dominio (*.irobot.uv.es*).
no_access	Determina que ordenadores no están autorizados a ejecutar este servicio en particular. El formato de especificación es igual al de *only_from*.
access_times	Indica el intervalo de horas en que el servicio esta disponible. El formato es hh:mm-hh:mm, donde hh va de 0 a 23 y mm de 0 a 59[14].
instances	Indica el número de servidores que pueden estar activos simultáneamente. El valor por defecto es sin límite.
per_source	Especifica el número de instancias permitidas de este servicio por dirección IP. Su valor es un entero o UNLIMITED si no se desea limitarlo.
cps	Especifica el número máximo de conexiones por segundo que pueden ser recibidas por este servicio. Sus argumentos son dos enteros, el primero indica el número máximo de conexiones que pueden ser recibidas y el segundo el intervalo en segundos en que el servicio estará deshabilitado si se sobrepasa el valor anterior.
deny_time	Especificá el tiempo de denegación de acceso a todos los servicios para una IP que ha sido indicada por el *flag* de SENSOR. Los valores posibles son FOREVER, NEVER y un valor numérico. FOREVER causa que la dirección IP no tenga acceso a los servicios hasta que *xinetd* sea restaurado, NEVER permite que la dirección IP continué teniendo acceso y el valor numérico indica el número de minutos en que le será denegado el acceso[15].

Opciones de limitación de uso de recursos:

Atributo	Descripción
nice	Determina la prioridad con la que se ejecuta el servidor.

[13] La dirección IP puede ser sustituida por el nombre del ordenador, en cuyo caso xinetd en el arranque determina la dirección IP de ese ordenador mediante el DNS.
[14] Toda petición del servicio es aceptada en ese intervalo de tiempo, no limitando en cualquier caso la duración del servicio, que puede exceder de dicho rango.
[15] Esta opción permite en muchos casos detener ataques de denegación de servicio desde una dirección IP.

Atributo	Descripción
max_load	Es un número en coma flotante que indica la carga (porcentaje de CPU) máxima para un servicio. En caso de que dicho valor se sobrepase el servicio dejará de aceptar conexiones.
rlimit_as	Determina el limite de memoria del servicio, el límite se especificá como un entero seguido de K (kilobytes) o M (megabytes) o UNLIMITED para indicar que no existe límite.
rlimit_cpu	Especifica el máximo número de segundos que el servidor puede utilizar de CPU. El límite se especifica como un entero o UNLIMITED si no existe.
rlimit_data	Especifica el tamaño máximo de los datos que el servidor puede utilizar. El límite se especifica como un entero indicando los bytes o UNLIMITED si no existe.
rlimit_rss	Especifica el tamaño máximo del programa que debe permanecer residente. Un tamaño pequeño hace a este servicio candidato a ser volcado a disco cuando la cantidad de memoria disponible es baja.. El tamaño se especificá como un entero que indica el número de bytes o UNLIMITED si no existe.
rlimit_stack	Especifica el tamaño máximo de la pila que el servidor puede utilizar. El límite se especifica como un entero indicando los bytes o UNLIMITED si no existe.

Opciones de información y log:

Atributo	Descripción
banner	Indica el nombre de un fichero que será mostrado en el ordenador remoto cuando una conexión a este servicio es solicitada.
banner_success	Indica el nombre de un fichero que será mostrado en el ordenador remoto cuando una conexión a este servicio es aceptada.
banner_fail	Indica el nombre de un fichero que será mostrado en el ordenador remoto cuanto una conexión a este servicio es rechazada.
log_type	Determina el tipo de log que utiliza el servicio, existen dos formas, SYSLOG y FILE. SYSLOG tiene la sintaxis SYSLOG syslog_facility [syslog_level] y especifica que el log será enviado al fichero de log del sistema con la facilidad especificada por syslog_facility (daemon, auth, authpriv, user, mail, lpr, new, uucp, ftp, local0-7) y el nivel especificado por syslog_level (emerg, alert, crit, err, warning, notice, info, debug), si el nivel no esta presente se asume info. Por su parte FILE tiene la sintaxis FILE file [soft_limit [hard_limit]] e indica que la salida será grabada en el fichero especificado por file, teniendo dicho fichero un limite soft y hard de forma similar a como sucede con los limites soft y hard en las cuotas de los usuarios.
log_on_success	Indica la información que será almacenada en el log cuando el servidor empieza y cuando termina. Puede ser cualquier combinación de los valores PID (identificador del proceso servidor), HOST (dirección del ordenador remoto), USERID (identificador del usuario), EXIT (código de terminación del servidor) y DURATION (duración del servicio).
log_on_failure	Indica la información que será almacenada cuando la petición es rechazada. Sus valores son una combinación de HOST (dirección del ordenador remoto), USERID (identificador del usuario) y ATTEMPT (guarda que ha sucedido un fallo, esta opción esta implícita en las dos anteriores).

No todas las propiedades anteriores deben ser especificadas para cada servicio. Solo las siguientes propiedades necesitan ser especificadas para cada servicio, tomando el resto de propiedades sus valores por defecto:

- *socket_type*.
- *user* (solo para servicios no internos).

- *server* (solo para servicios no internos).
- *wait*.
- *protocol* (solo para servicios RPC o no listados).
- *rpc_version* (solo para servicios RPC).
- *rpc_number* (solo para servicios RPC no listados).
- *port* (solo para servicios no listados).

Además, el fichero de configuración */etc/xinetd.conf* puede contener, además, una sola entrada con los atributos asignados por defecto a todos los servicios[16]. El formato de dicha entrada es:

```
defaults
{
      <atributo> = <valor>...
      ...
}
```

Donde el campo *atributo* puede tomar los valores:

- *log_type*.
- *bind*.
- *per_source*.
- *umask*.
- *log_on_success*.
- *log_on_failure*.
- *only_from*.
- *passenv*.
- *instances*.
- *disable*.
- *enabled*.
- *banner*.
- *banner_success*.
- *banner_fail*.
- *per_source*.
- *groups*.
- *cps*.
- *max_load*.

Un ejemplo de fichero */etc/xinetd.conf* es el siguiente:

```
defaults
{
      log_type        = SYSLOG daemon info
      log_on_failure  = HOST
      log_on_success  = PID HOST DURATION EXIT
      cps             = 50 10
      instances       = 50
```

[16] Estos atributos serán modificados si existe la entrada correspondiente en la configuración del servicio, serán modificados de la forma que corresponda a cada uno de ellos.

```
      per_source      = 10
      v6_only         = no
      groups          = yes
      umask           = 002
}

includedir /etc/xinetd.d
```

En este fichero podemos ver que se indica que por defecto un servicio puede tener 50 instancias como máximo a la vez con un máximo de 10 instancias por dirección IP, que el tipo de log es SYSLOG utilizando la facilidad *daemon*. Además, indicamos que la información en caso de aceptar una conexión será la IP del ordenador remoto, el PID del proceso servidor, etc., y en caso de rechazar la solicitud de conexión la IP del ordenador remoto. También especificamos que a cada servidor se le permite que se le efectúen 50 solicitudes por segundo y, una vez sobrepasado ese límite, el servicio quedará desactivado durante 10 segundos. Por último, especificamos que pueda utilizar los grupos de los servicios que ejecuta y que las mascara por defecto es 002.

Al final del fichero podemos ver que aparece una línea *includedir /etc/xinetd.d* que indica que se incluya como configuración de *xinetd* la información contenida en los ficheros de ese directorio[17]. Algunos ejemplos de ficheros existentes en el directorio */etc/xinetd.d*, así como de su configuración, pueden verse a continuación[18]:

```
# Fichero daytime-stream.
# Servicio interno TCP que devuelve el día y la hora del
# sistema.
service daytime
{
     disable         = yes
     id              = daytime-stream
     type            = INTERNAL
     wait            = no
     socket_type     = stream
}

# Fichero daytime-dgram.
# Servicio interno UDP que devuelve el día y la hora del
# sistema.
service daytime
{
     disable         = yes
     id              = daytime-dgram
     type            = INTERNAL
     wait            = yes
     socket_type     = dgram
}

# Fichero rsync
# Sincroniza de forma remota archivos y directorios.
service rsync
{
```

[17] Téngase en cuenta la excepción que sobre ciertos ficheros establece *includedir*.

[18] Para disminuir el tamaño del texto aquí mostrado, se han eliminado todas aquellas líneas que aparecen comentadas en los ficheros.

```
      disable         = yes
      flags           = IPv6
      socket_type     = stream
      wait            = no
      user            = root
      server          = /usr/bin/rsync
      server_args     = --daemon
      log_on_failure  += USERID
}

# Fichero telnet.
# Servidor de telnet que utiliza la transmisión de usuarios y
# contraseñas sin cifrado para la autentificación.
service telnet
{
      flags           = REUSE[19]
      socket_type     = stream
      wait            = no
      user            = root
      server          = /usr/sbin/in.telnetd
      log_on_failure  += USERID
      disable         = yes
}

# Fichero tftp
# Servidor de tftp utilizado para arranque remoto de sistema,
# descarga de ficheros de configuracion, etc.
service tftp
{
      socket_type     = dgram
      protocol        = udp
      wait            = yes
      user            = root
      server          = /usr/sbin/in.tftpd
      server_args     = -s /var/lib/tftpboot
      disable         = yes
      per_source      = 11
      cps             = 100 2
      flags           = IPv4
}
```

Analizando los ficheros podemos ver que en los servicios de *daytime* hemos definido identificadores (atributo *id*) para los servicios (*daytime-stream* y *daytime-dgram*), pues existen servicios de igual nombre con versiones para TCP y UDP.

También podemos ver en los ejemplos como en el servicio de *rsync* utilizamos la opción *server_args* para indicarle que argumentos han de pasarse al servidor en el momento de ser lanzado. Por último, podemos ver que el servidor de *tftp* posee la opción *cps = 100 2*, lo cual le indica que admita como máximo 100 conexiones simultáneas y si este número se sobrepasa este 2 segundos desactivado antes de comprobar si puede aceptar nuevas conexiones, sobrescribiendo el valor por defecto de la opción que se especifica en la sección *defaults* del fichero */etc/xinetd.conf*.

[19] La bandera *REUSE* es obsoleta, pues en la actualidad todos los servicios tienen implícitamente la bandera *REUSE* activada.

Configuración del envolvente de acceso de Linux.

Antes de explicar la configuración del envolvente de acceso de Linux, es conviene aclarar la interacción que posee el envolvente de acceso con el envolvente de acceso que implementa el servidor *xinetd*.

En un sistema que ejecuta el envolvente de acceso de Linux, este envolvente de acceso examina todas las peticiones de servicios que son realizadas, de forma que solo son admitidas las que su configuración indique. Si, con posterioridad, esa petición de servicio es procesada por el servidor *xinetd*, este aplicará las reglas que tenga implementadas para dicho servicio, por lo que, cualquier petición de un servicio que este prohibido en el envolvente de acceso, nunca llegará a ser examinada por el envolvente de acceso de *xinetd*.

Introduciéndonos ya en este punto, y como explicamos, los envolventes de acceso nos permiten controlar y limitar el acceso a los servicios que ejecuta un ordenador. Para ello, los envolventes de acceso se basan en dos ficheros, */etc/hosts.allow* y */etc/hosts.deny* que especifican respectivamente los servicios a los que se permite acceso y desde que ordenadores y los servicios a los que se deniega el acceso y desde que ordenadores[20].

Las reglas de funcionamiento de los envolventes en cuanto al tratamiento de dichos ficheros ya fueron explicadas con anterioridad cuando se trato del origen de los envolventes de acceso. En este punto veremos la sintaxis de esos ficheros, etc., así como un ejemplo de los mismos.

La sintaxis de las reglas de acceso es la siguiente:

```
<lista de servicios>: <lista de clientes> [: spawn <comando de shell>]
```

Donde <*lista de servicios*> es una lista de uno o más servicios separados por espacios, donde los servicios se especifican mediante el nombre del servidor que los proporciona (*vsftpd*, *sshd*, *in.telnetd*, etc.) o el nombre interno del servicio si lo proporciona xinetd (*daytime*, *echo*, etc.), <*lista de clientes*> es uno o más nombres de ordenador, direcciones IP o patrones separados por espacios; y <*comando de shell*> es opcional e indica una acción a ejecutar si una regla se cumple.

La lista de servicios o la lista de clientes pueden ser simplificadas utilizando patrones, pues permiten especificar grupos de servicios o grupos de clientes de forma sencilla.

El patrón más utilizado es el carácter punto (.) al principio de una cadena de caracteres o al final de una especificación de direcciones IP. Así, si escribimos *.irobot.uv.es*[21], nos estamos refiriendo a todo ordenador del dominio del Instituto de

[20] El envolvente de acceso general limita también los programas que pueden ejecutar xinetd, con lo cual debe configurarse teniendo en cuenta esta doble limitación.

[21] Siempre que sea posible, es conveniente utilizar las direcciones IP en vez de los nombres, pues para este último caso es necesario usar un DNS que puede ser objeto de modificación por parte de un intruso antes de intentar el acceso a un servicio de otro ordenador.

Robótica, mientras que si escribimos *147.156.* nos referimos a todo ordenador cuya dirección IP comienza por 147.156, esto es, ordenadores de la Universidad de Valencia. Además del carácter punto pueden utilizarse como comodines los caracteres asterisco (*) e interrogación (?), con las mismas funcionalidades que poseen en el interprete de comandos del sistema operativo[22].

Además de los caracteres anteriores, existen unas palabras clave que pueden ser usadas en las reglas de acceso en lugar de especificar la lista de clientes. Estas palabras clave pueden verse en la tabla siguiente:

Palabra	Descripción
ALL	Especifica todos los ordenadores.
LOCAL	Especifica todos los ordenadores de nuestra red local, esto es, que no contienen el carácter '.' en su nombre.
KNOWN	Especifica todos los ordenadores cuyo nombre o dirección IP son conocidos.
UNKNOW	Especifica todos los ordenadores cuyo nombre o dirección IP es desconocidos.
PARANOID	Especifica todos los ordenadores cuyo nombre no corresponde con su dirección IP.

Las palabras KNOWN, UNKNOW y PARANOID deben ser utilizadas con precaución pues un error o alteración del DNS puede producir que ordenadores o usuarios no autorizados obtengan acceso a los servicios.

La <*lista de clientes*> puede contener un operador, el operador EXCEPT, que permite combinar dos listas en la misma línea. Cuando EXCEPT es utilizada entre dos listas, la <*lista de servicios*> se aplica a todas las entradas contenidas en la primera lista excepto a aquellas contenidas en la segunda lista. Para entender mejor el uso de estas palabras clave consideremos el siguiente ejemplo de línea en el fichero *hosts.allow*:

```
vsftpd: .irobot.uv.es EXCEPT amparo.irobot.uv.es glup.irobot.uv.es
```

En dicha línea estamos indicando que permitimos el uso del servicio de FTP proporcionado por *vsftpd* a todos los ordenadores del dominio del Instituto de Robótica excepto a los ordenadores de nombre *amparo* y *glup*[23].

Además de lo expuesto con anterioridad, la palabra clave ALL puede ser también utilizada en la <*lista de servicios*>, significando todos los servicios. Su uso podemos verlo en el siguiente ejemplo, que muestra además un uso más complicado del operador EXCEPT:

```
ALL: ALL EXCEPT in.telnetd: amparo.irobot.uv.es
```

Que permite a todos los ordenadores utilizar todos los servicios excepto al ordenador *amparo* al que no autoriza a utilizar el servicio de telnet (que es servido por el ejecutable de nombre *in.telnetd*).

[22] Esto es, el carácter comodín * sustituye a cero o más caracteres, mientras que el carácter comodín ? sustituye solo a un carácter que además debe estar presente.

[23] Es necesario tener en cuenta que la no autorización expresa a esos dos ordenadores no implica la denegación de utilización de ese servicio, esto debería hacerse constar de forma implícita en el fichero /etc/hosts.deny.

Por último, la opción *spawn* permite ejecutar un comando en caso de que sea cierta una regla. Así, por ejemplo, podemos escribir en un fichero información sobre el ordenador, etc., al que se autoriza o deniega el uso de un servicio. Un ejemplo de regla que permite esto es:

```
in.telnetd: .irobot.uv.es : spawn (/bin/echo `date` %c >> /var/log/telnet.log) &
```

Que escribiría la hora (ejecución del comando date) y la dirección IP y usuario del ordenador que intento acceder al servicio de *telnet*.

Como ha podido verse en el ejemplo anterior, existen una serie de caracteres que indican que debe escribirse información obtenida a través de la conexión o intento de conexión. Estos caracteres se encuentran en la tabla siguiente:

Carácter	Descripción
%a	La dirección IP del cliente.
%A	La dirección IP del servidor.
%c	Proporciona una variedad de información como el nombre del usuario y el nombre del ordenador, o el nombre del usuario y la dirección IP.
%d	El nombre del servicio solicitado.
%h	El nombre del cliente (o dirección IP si el nombre no existe).
%H	El nombre del servidor (o dirección IP si el nombre no existe).
%n	El nombre del cliente. Si no existe se escribe *unknow*. Si el nombre del cliente y su dirección IP no coinciden se escribe *paranoid*.
%N	El nombre del servidor. Si no existe se escribe *unknow*. Si el nombre del cliente y su dirección IP no coinciden se escribe *paranoid*.
%p	El identificador del proceso del servicio.
%s	Proporciona una variedad de información como el identificador del proceso y el nombre o dirección IP del servidor.
%u	El nombre del cliente. Si no existe se escribe *unknow*.

Un ejemplo de ficheros *hosts.allow* y *hosts.deny* es el siguiente:

```
# Fichero hosts.allow
# Servicio FTP (21/tcp) permitido a todo el mundo
vsftpd: ALL
# Servicio SSH (22/tcp) permitido a todo el mundo
sshd: ALL
# Servicio daytime (13/tcp) permitido solo a Robótica
daytime: .irobot.uv.es

# Fichero hosts.deny
ALL : ALL : spawn (/bin/echo `date` %h %d >> /var/log/deny.log) &
```

Como puede verse, el fichero *hosts.allow* permite explícitamente el servicio de FTP y SSH a todos los ordenadores de Internet, mientras que limita el uso del servicio de *daytime* a los ordenadores del dominio *.irobot.uv.es*.

Por su parte, el fichero *hosts.deny* niega todos los servicios a todos los ordenadores, de forma que si un ordenador no se encuentra entre los autorizados de forma explícita en *hosts.allow*, su acceso a cualquier servicio será denegado[24]. La

[24] Una política de seguridad tan restrictiva permite controlar la seguridad del sistema y evitar posibles problemas por errores en la configuración.

denegación incluye la escritura en un fichero de log de la fecha, el nombre o dirección IP del cliente que solicito el servicio y el nombre del servicio solicitado.

Por último, volver a comentar que aunque un servicio sea permitido en la configuración del envolvente de acceso, este puede ser denegado por el fichero de configuración del propio servicio o bien, en el caso de servicios que son ejecutados mediante el servidor *xinetd*, en el fichero de configuración del servidor *xinetd*.

Ejercicios.

1- Configurar el servidor de xinetd para que permita el acceso al servicio de telnet a un máximo de 4 ordenadores de forma simultánea y solo durante la franja horaria de 08:00 a 20:00 horas.

2- Configurar el servidor de xinetd para que permita el acceso al servicio de daytime bajo TCP para los ordenadores 147.156.222.34 y 147.156.222.65, pero limitando el número máximo de conexiones por segundo a 1, sobrepasado el cual el servicio se debe deshabilitar durante un minuto.

3- Configurar los envolventes de acceso para que permitan el acceso a todos los ordenadores de Internet únicamente a los servicios de FTP y SSH, denegando el acceso al resto de servicios.

4- Configurar los envolventes de acceso de forma que se permita el acceso al servicio de FTP a todos los ordenadores de Internet, y al servicio de SSH a todos los ordenadores de Robótica excepto el ordenador amparo.irobot.uv.es. Además, deseamos que en caso de que sea denegado el acceso a algún servicio se escriba en un fichero de log el nombre del servicio, la hora y la dirección IP del ordenador que solicito el servicio.

5- Configurar tanto los envolventes de acceso como el servidor de xinetd de forma que se permita el acceso al servicio de daytime bajo TCP y UDP solamente a los ordenadores de la universidad (147.156.), el servicio de SSH a todos los ordenadores de Internet y el servicio de FTP a todos los ordenadores de Internet excepto los ordenadores de la universidad. En cualquier caso, cuando se permita o deniegue el acceso se debe escribir en un fichero información sobre el mismo.