Elvira Mifsud -k idatzia
Asteartea, 2009(e)ko abuztua(r)en 25-(e)an 09:15etan

There are no translations available.
De esta forma profesor y alumnos pueden compartir directorios, con las restricciones adecuadas, y pueden intercambiar archivos en el aula informática.

1 Introducción

En un entorno de aula informática se hace imprescindible disponer de un servicio que permita el acceso seguro a archivos remotos de forma trasparente. Tanto el profesor como los alumnos, en determinadas circunstancias, necesitan disponer de esta facilidad de intercambio de información que garantice la seguridad y confidencialidad de la misma.

NFS proporciona este servicio siguiendo la estructura cliente-servidor. El servidor NFS comparte una serie de directorios seleccionados con unas condiciones de seguridad concretas. El cliente NFS, si está autorizado para ello, puede 'montar' dichos directorios en su propio sistema de archivos pudiendo acceder a los archivos como si fueran locales. El montaje lo puede realizar en secuencia de arranque del equipo o cuando lo necesite.

De esta forma profesor y alumnos pueden compartir directorios, con las restricciones adecuadas, y pueden intercambiar archivos dentro de la red de área local configurada en el aula informática.

Esta forma de trabajar es válida para entornos Unix/Linux. De momento NFS no permite la interoperabilidad con determinados sistemas de archivos Windows. Para poder trabajar con ciertos sistemas de archivos de red en plataformas mixtas Windows/Linux se ha de utilizar el antiguo protocolo SMB, hoy llamado CIFS.

En la actualidad las versiones de la distribución Ubuntu soportan la conexión con equipos Windows directamente utilizando el protocolo SMB. No requiere prácticamente de ninguna configuración adicional y está disponible en la opción de menú *Lugares > Conectar con el servidor* . Este menú permite seleccionar el *Tipo de servicio > Compartido por Windows* .

2 ¿Qué es NFS?

Las siglas NFS significan Sistema de Archivos de Red (del inglés Network File System) y fue

desarrollado por SUN Microsystems en 1984. Su función en una red es permitir que un equipo GNU/Linux pueda montar y trabajar con un sistema de archivos de otro equipo de la red como si fuera local.

Cuando hablamos de sistema de archivos nos estamos refiriendo a las diferentes formas de que disponen los sistemas operativos de estructurar su información sobre los dispositivos físicos. Por ejemplo, en GNU/Linux es normal hablar de sistemas de archivos ext3, XFS, ReiserFS. En Windows son sistemas de archivos típicos fat16, fat32, NTFS, etc. En el artículo hablaremos tanto de sistemas de archivos compartidos como de directorios compartidos. En realidad los sistemas de archivos utilizan los directorios para organizar los archivos.

En este sentido NFS no es realmente un sistema de archivos físico, sino que constituye una capa de abstracción que, aplicada sobre cualquier sistema de archivos físico, permite su utilización de forma remota por otros equipos/usuarios.

El servicio NFS utiliza las llamadas a procedimientos remotos basadas en el protocolo RPC (del inglés, Remote Procedure Call) que permite desde un equipo (cliente) ejecutar código ubicado en otro equipo remoto (servidor) mediante el establecimiento de sockets (IP+puerto) entre ambas.

Aunque al servicio se le suele conocer con el nombre NFS, realmente NFS es un protocolo de nivel de Aplicación
y por debajo, el protocolo subyacente que utiliza NFS son las
Llamadas a Procedimientos Remotos (RPC) de nivel de Sesión, también
utiliza TCP/
UDP en el nivel Transporte e IP en el nivel de Red.

NFS es un protocolo sin memoria (state-less) en algunas de sus versiones. Es decir, el servidor no recuerda las solicitudes anteriores. Por tanto, cada llamada a un procedimiento contiene toda la información necesaria para su finalización. Si el servidor NFS falla, el sistema cliente repetirá las solicitudes de NFS hasta que obtenga una respuesta. Además, el servidor no realiza tareas de recuperación frente a fallos.

3 ¿Cuándo necesitamos NFS?

Realmente los diferentes escenarios en los que se hace necesaria la compartición de archivos es muy amplia. Por ejemplo y dentro del ámbito del aula:

1.

El profesor quiere compartir con sus alumnos en modo lectura los ejercicios que deben realizar sus alumnos. NFS puede incluir los archivos correspondientes en un directorio que exportará con permiso de lectura y al que los alumnos podrán acceder.

2.

Un grupo de alumnos está realizando un trabajo y deben compartir una serie de archivos y trabajar sobre ellos. NFS permite crear un directorio en el servidor y se exporta a todas las máquinas/usuarios que colaboran en el trabajo.

3.

En un entorno de aula informática en la que se trabaja con usuarios de red y se quieren tener centralizados los directorios home de todos ellos. NFS permite exportar y montar estos directorios /home de cada alumno de forma transparente. De esta forma se pueden controlar los accesos de los alumnos, la información almacenada en sus directorios de trabajo, la salvaguarda de dicha información,...

4.

Estamos en un entorno de aula en el que las máquinas de los alumnos disponen de poca capacidad y necesitan trabajar con aplicaciones que 'no caben' localmente. NFS permite exportar los directorios que contienen estas aplicaciones desde el servidor y los alumnos podrán ejecutarlas en sus máquinas.

5.

El administrador del aula necesita que en su aula todas las máquinas tengan el mismo software y con idéntica configuración. NFS permite exportar del servidor el directorio que contiene el software requerido (por ejemplo, /usr) y el directorio que contiene las configuraciones correspondientes (por ejemplo, /etc). Esto mismo puede servir para hacer instalaciones completas de sistemas operativos por la red si los equipos carecen de unidades de CD-ROM/DVD.

4 Versiones de NFS

Las versiones de NFS mas importantes son NFSv2 (RFC 1094), NFSv3 (RFC 1813) y NFSv4 (RFC 3530).

La versión 2 de NFS es la más extendida y soportada por los sistemas operativos, también es la mas antigua e insegura. La versión 3 es mas potente pero no es completamente compatible con clientes NFSv2. Ambas versiones pueden trabajar tanto con TCP como UDP como protocolo de transporte creando conexiones de red entre el cliente y el servidor sin supervisión (state-less). La ventaja de utilizar UDP es que, al ser una conexión desatendida, se minimiza el tráfico de red, pero si el servidor NFS cayera por cualquier circunstancia, los clientes NFS seguirían enviando peticiones al servidor produciendo el efecto contrario, que es la saturación de la red.

En general las versiones 2 y 3 de NFS permiten controlar la exportación y montaje de sistemas de archivos en función del equipo que hace la solicitud, pero no del usuario. Es decir no se contempla un control de acceso al sistema de archivos por usuario. Sólo para los equipos. Esto implica que si un sistema de archivos es exportado desde el servidor NFS, cualquier usuario de un equipo remoto cliente NFS podría acceder a él. Los únicos mecanismos de seguridad que quedan en este caso son los permisos de acceso (sólo lectura) o utilizar un usuario y grupo únicamente. Lógicamente esto limita bastante la idea de compartición que tenemos todos.

En el caso de la versión 4 de NFS (http://www.nfsv4.org) estos problemas de seguridad desaparecen pero, a cambio, tiene unos requerimientos de configuración y servicios adicionales mucho mas importantes. Por ejemplo, en la versión 4 la utilización de mecanismos para la autenticación de los usuarios es obligatoria. Para ello y en función del tipo de seguridad seleccionada, se requiere la utilización del servicio Kerberos cuya misión será funcionar como servidor de entrega de tickets (KDC) y que debe estar configurado y funcionando correctamente antes de configurar el servidor NFSv4. Este requerimiento proporciona seguridad al servicio NFS a cambio de incluir mayor complejidad a su configuración y puesta a punto.

Otra característica importante de NFS4 es la utilización de ACLs (Listas de Control de Acceso) al estilo Windows y que no son soportadas por las versiones 2 y 3 de NFS. Cuando hablamos de ACLs nos referimos a los permisos o derechos de acceso que tiene cada usuario sobre un archivo o directorio y que vienen especificados a modo de listas editables por el administrador

del sistema.

En este artículo trataremos la versión 3, pondremos de relieve las diferencias entre versiones y al final del artículo daremos una aproximación a lo que sería la configuración de un servidor y cliente NFSv4 utilizando la seguridad básica del sistema.

5 ¿Cómo funciona NFS?

Vamos a suponer que un cliente NFS (equipo) intenta montar un directorio, exportado desde el servidor NFS, en un directorio local. Para ello se utiliza la orden mount indicando el tipo de sistema de archivos (-t) nfs, la máquina remota (donde está instalado y configurado el servidor NFS) y el directorio a montar y el punto de montaje local. El directorio local debe existir previamente.

```
$sudo mount -t nfs maquina_remota:/home /dir_local
```

La orden mount intenta conectar con el demonio rpc.mountd, que está ejecutándose en la máquina remota, vía RPC. El servidor NFS comprueba si la máquina que hace la petición de montaje (cliente NFS) tiene permisos sobre el directorio /home. Si los tiene se lleva a cabo el montaje como si se tratase de un dispositivo físico (disco duro, CD/DVD, USB, etc). Cuando se acceda al directorio /dir_local desde la máquina cliente se estará accediendo al directorio /home de la máquina remota.

Cuando el directorio /dir_local tenga ya montado el directorio /home de la máquina remota, la única protección que tienen los archivos de dicho directorio son sus permisos.

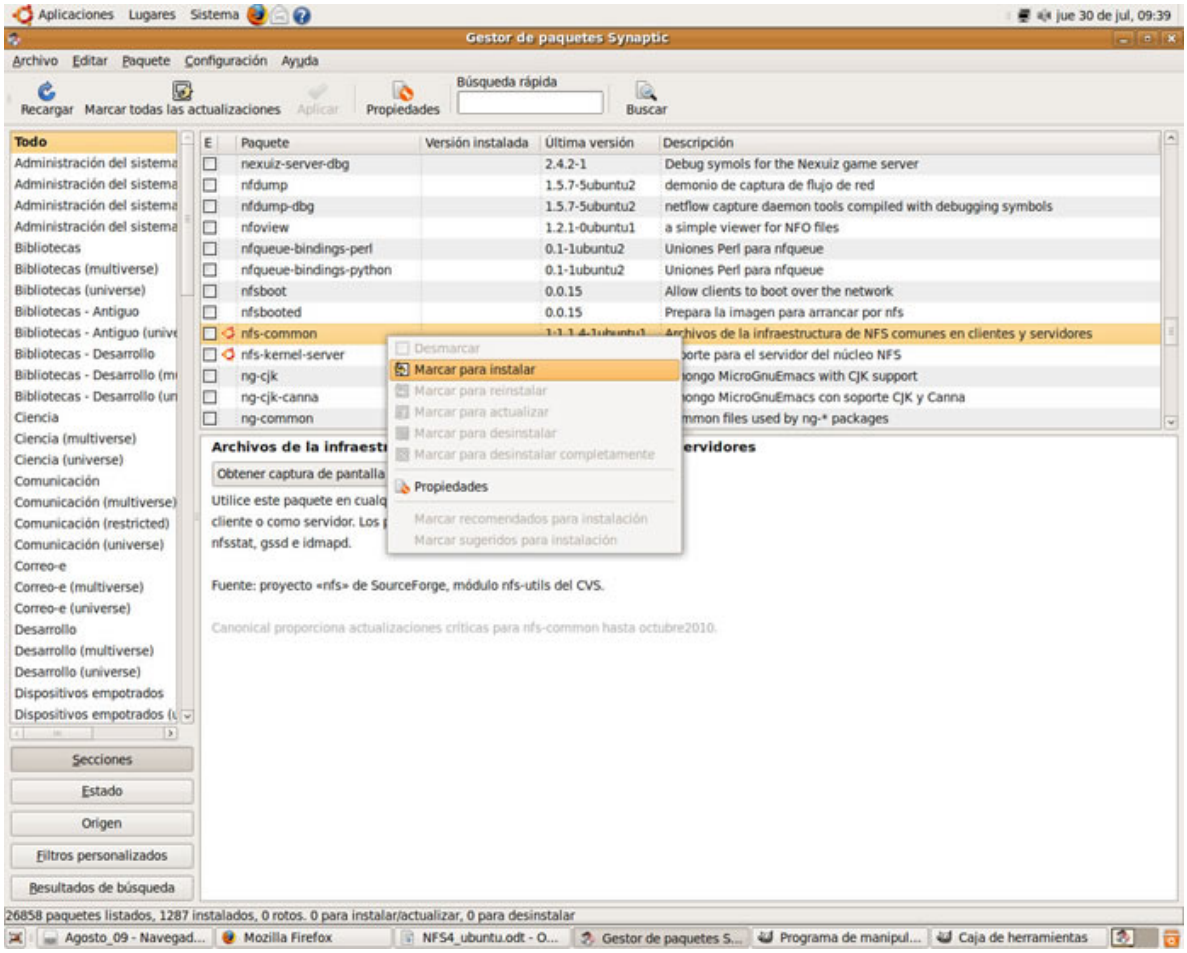
Si un usuario accede a un archivo dentro del directorio NFS se genera una llamada RPC al demonio rpc.nfsd en el servidor. En esta llamada van incluidos como parámetros el descriptor del archivo al que se intenta acceder y el UID y GID del usuario. Con estos valores se comprueban los derechos de acceso sobre el archivo requerido. La situación perfecta es que estos identificadores coincidan tanto en el cliente como en el servidor.

6 Instalación y configuración de NFSv3

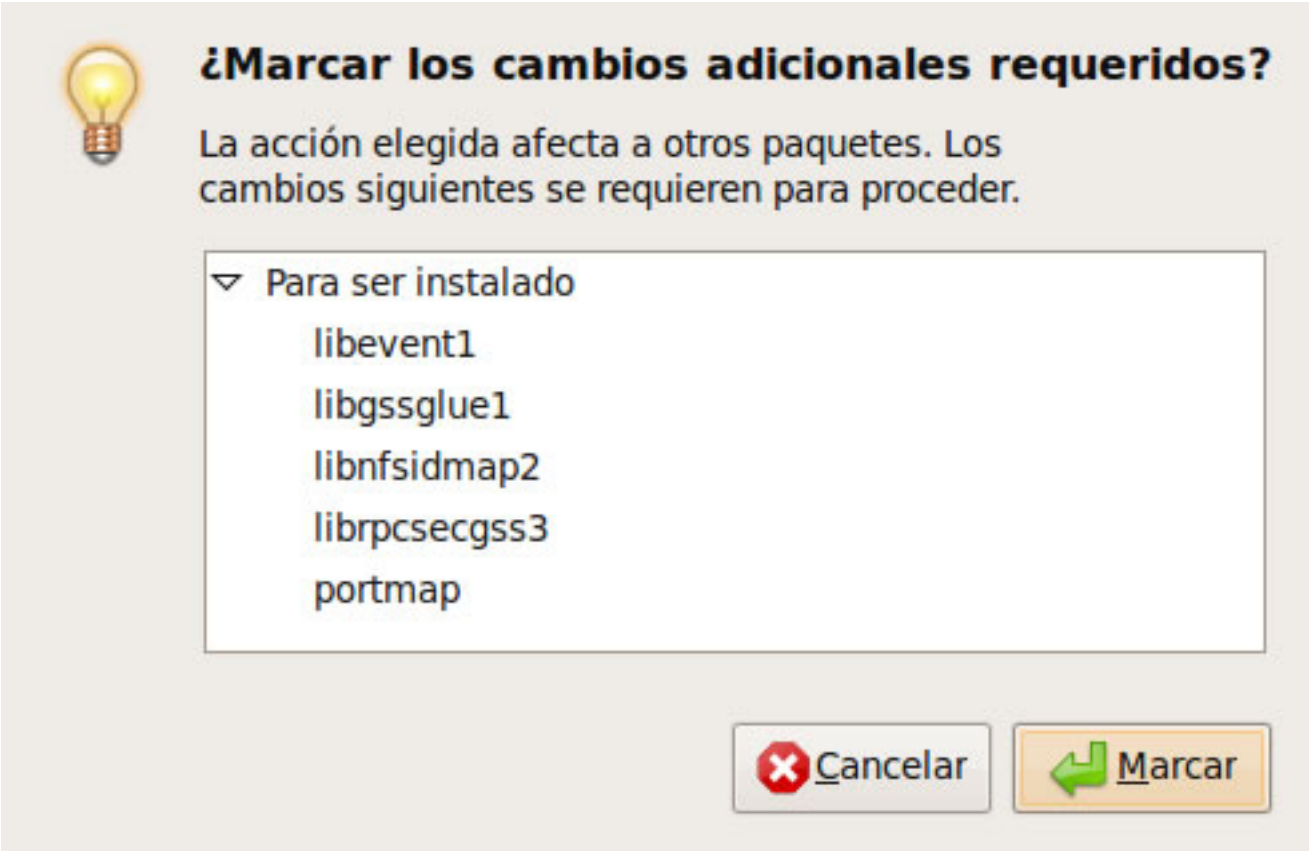
6.1 Servidor NFS

Para instalar el servidor NFS en primer lugar habrá que actualizar el sistema. Par ello ir a *Sistema > Administrador > Synaptic*
y actualizar el sistema pulsando el botón
Recargar
.

Ya actualizado buscar entradas nfs y marcar para instalar el paquete **nfs-common**.



Comprobar que se van a instalar, además del paquete seleccionado, otros paquetes arrastrados por él.



De la misma forma instalar el paquete **nfs-kernel-server**.

Podemos comprobar que el kernel tiene soporte para nfs ejecutando la orden:

```
$ sudo cat /proc/filesystems |grep nfs
```

```
nodev nfsd
```

También podemos comprobar si el servicio está ya activo en el sistema de la forma siguiente:

```
$ sudo rpcinfo -p

programa versus puerto del protocolo

100000 2 tcp 111 portmapper

100000 2 udp 111 portmapper

100024 1 udp 52944 status

100024 1 tcp 60149 status

100003 2 udp 2049 nfs

 .....

100021 1 udp 56598 nlockmgr

 .....

100021 1 tcp 58434 nlockmgr
```

.....

100003 2 tcp 2049 nfs

.....

100005 1 udp 37578 mountd

100005 1 tcp 48759 mountd

.....

Esta orden muestra los servicios activos basados en RPC con el número de puerto, número de programa RPC, versión y tipo de protocolo (TCP o UDP).

Para comprobar si existen las entradas nfs y mountd así como el portmap como procesos del sistema ejecutándose:

$ sudo **ps aux | grep portmap**

daemon 4693 0.0 0.0 1940 544 ? Ss 09:41 0:00 /sbin/portmap

$ sudo ps aux | grep mountd

root 5094 0.0 0.0 2120 316 ? Ss 09:42 0:00 /usr/sbin/rpc.mountd -manage-gids

$ sudo ps aux | grep nfs

root 5082 0.0 0.0 0 0 ? S< 09:42 0:00 [nfsd4]

root 5083 0.0 0.0 0 0 ? S< 09:42 0:00 [nfsd]

root 5084 0.0 0.0 0 0 ? S< 09:42 0:00 [nfsd]

root 5085 0.0 0.0 0 0 ? S< 09:42 0:00 [nfsd]

Como se comprueba, la instalación de los paquetes implicados en el servicio NFS nos ha dejado una versión inicial del servidor operativa.

La versión descargada para Ubuntu Jaunty es configurable como NFSv3 o NFSv4.

6.2 Demonios del servicio NFS

Los demonios imprescindibles del servicio NFS son los siguientes:

-

rpc.mountd: demonio para el montaje remoto. Se ejecuta en el servidor. Recibe la petición de montaje desde un cliente NFS y comprueba en el archivo /var/lib/nfs/xtab si el sistema de archivos está exportado. Si el sistema de archivos está disponible, permite las solicitudes de acceso de NFS y después proporciona información sobre los sistemas de archivos mediante el comando showmount. Comprueba también que el cliente tenga permiso para solicitar acceso.

-

rpc.nfsd: demonio para servir archivos. Gestiona las solicitudes del cliente una vez mountd ha dado el visto bueno al cliente. Se pueden arrancar varias copias de este demonio. Utiliza el puerto TCP/UDP 2049.

-

rpc.portmap: es el encargado de decir a los clientes donde está localizado (número de puerto) el servicio real en el servidor. Como los servicios basados en RPC utilizan portmap para atender las peticiones de los clientes, este servicio debe estar disponible antes de cualquier otro servicio o demonio de NFS. No se utiliza en NFSv4. Utiliza el puerto TCP/UDP 111. Para comprobar que está activo ejecutar la orden:

```
$ sudo portmap status
```

-

rpc.lockd: encargado de proporcionar el servicio de bloqueo de archivos para asegurar su consistencia ya que pueden ser accedidos de forma concurrente. Se ejecuta tanto en el servidor como en el cliente.

-

rpc.statd: trabaja conjuntamente con lockd para permitir la recuperación en caída de sistemas. Mantiene información sobre los procesos en los clientes que poseen locks de archivos de determinado servidor. Cuando el servidor NFS se recupera statd informa a los otros procesos statd de los clientes, que el servidor se ha recuperado, y así ellos intentarán resolver los locks que tenían anteriormente. En los clientes statd se utiliza para avisar al servidor de que el cliente ha caído y así poder liberar los archivos que tuviera ese cliente bloqueados.

Los demonios estarán escuchando es sus puertos correspondientes. Podemos comprobarlo ejecutando la orden:

```
$ sudo netstat -tunpl

tcp 0 0 0.0.0.0:2049 0.0.0.0:* ESCUCHAR -

tcp 0 0 0.0.0.0:111 0.0.0.0:* ESCUCHAR -

udp 0 0 0.0.0.0:2049 0.0.0.0:* -

udp 0 0 0.0.0.0:111 0.0.0.0:* -

.....
```

6.3 Arranque y parada del servicio NFS

-

Iniciar el servicio NFS:

```
$ sudo /etc/init.d/nfs-kernel-server start
```

-

Detener el servicio NFS:

```
$ sudo /etc/init.d/nfs-kernel-server stop
```

-

Reiniciar, es decir, parar e iniciar el servicio NFS:

```
$ sudo /etc/init.d/nfs-kernel-server restart
```

```
* Stopping NFS kernel daemon [ OK ]

* Unexporting directories for NFS kernel daemon... [ OK ]

* Exporting directories for NFS kernel daemon... [ OK ]

* Starting NFS kernel daemon [ OK ]
```

-

Forzar a recargar los archivos de configuración del servicio NFS sin parar el servidor:

```
$ sudo /etc/init.d/nfs-kernel-server reload
```

Hay que recordar que, para que el servicio NFS funcione, debe estar ejecutándose

previamente el demonio portmap.

6.4 Archivos de configuración

Los archivos de configuración del servicio NFS son los siguientes:

-

/**etc**/**fstab:** contiene los sistemas de archivos que pueden ser montados desde sistemas remotos en secuencia de arranque del equipo.

-

/**etc**/**exports:** contiene una lista de los directorios del sistema local que se van a exportar a sistemas remotos utilizando NFS y los permisos de uso. La existencia de este archivo determina si el sistema local es un servidor de NFS. Este archivo contiene una línea por cada directorio a compartir.

-

/**var**/**lib**/**nfs**/**etab:** contiene una lista de los sistemas de archivos actualmente exportados para el sistema local. Esta información es actualizada en este archivo cuando se ejecuta el comando exportfs que lee el archivo /etc/exports.

-

/**etc**/**hosts.allow** y /**etc**/**hosts.deny:** NFS utiliza estos archivos para comprobar a qué máquinas se les acepta o deniega el uso de NFS. En general este sistema de comprobación se suele conocer con el nombre de wrappers TCP

6.5 Exportación de un directorio

La estructura de las líneas del archivo /etc/exports es la siguiente:

```
directorio equipo1(opcion11,...) equipo2(opcion21,...)
```

Donde:

directorio*:* es el nombre del directorio que se comparte.
E**quipo***X:* son los clientes NFS que tendrán acceso al directorio compartido. Estos equipos se pueden identificar mediante su dirección IP o su nombre DNS (si se tiene disponible un servidor DNS). Se admite la utilización de los comodines '*' y '?', aunque su utilización puede ser algo peligrosa sino se conoce bien como se producirá su expansión.

option*XY:* son las diferentes opciones que asignamos a este directorio para ese equipo en concreto y que determinarán los privilegios de acceso a él. De todas la opciones disponibles, las mas significativas son:

-

ro|*rw*: el directorio será compartido en solo lectura (*ro*) y es la opción por defecto. El directorio será compartido en lectura y escritura (*rw*).

-

sync|*async*: *sync* comunica al usuario los cambios realizados sobre los archivos cuando realmente se han ejecutado y es la opción recomendada. La opción *async* mejora el rendimiento y agiliza el funcionamiento del servicio, pero puede generar archivos corruptos si se produce algún tipo de fallo en el servidor.

-

no_subtree_check*:* permite que no se compruebe el camino hasta el directorio que se exporta, en el caso de que el usuario no tenga permisos sobre el directorio exportado.

-

root_squash | *no_root_squash* | *all_squash*

-

root_squash indica que un usuario identificado como *root* tendrá acceso al directorio

compartido sólo con privilegios de usuario anónimo. De esta forma se ha degradado al root al usuario local de privilegios mas bajos protegiendo así los archivos en el servidor NFS. Esta opción se conoce también con el nombre de 'aplastamiento del root'. Para el resto de usuarios se intenta conservar su UID y GID en el servidor.

-

no_root_squash desactiva la opción anterior, es decir, los accesos realizados como root desde el cliente serán también de root en el servidor NFS.

-

all_squash indica que todos los clientes, incluido root, tendrán acceso al directorio con privilegios de un usuario anónimo. No se mantienen los UID y GID de ningún usuario.

-

Si se utiliza alguna de las opciones squash podemos indicar cuál el el UID y GID del usuario con el que se quiere que se acceda, en lugar del anónimo. En este caso hemos de indicar a continuación de la opción squash lo siguiente:

(rw,all_squash,anonuid=1002,anongid=1002)

Y significa que la conexión del cliente NFS se hará con los UID y GID 1002.

El primer paso ahora es decidir qué directorio se va a compartir y qué equipos van a poder acceder al mismo. En nuestro caso vamos a compartir el directorio /alumnos a los equipos de la red 192.168.100.0. Para ello editamos el archivo /etc/exports con gedit[1] (*Aplicaciones > Accesorios > Editor de texto*
) y escribimos lo siguiente:

```
/alumnos 192.168.100.0/24(rw,sync,all_squash)
```

Con esta línea, incluida en el archivo /etc/exports, lo que se está compartiendo es el directorio /alumnos a todas las máquinas de la red 192.168.100.0 en lectura y escritura (rw), se comunica al usuario los cambios realizados sobre los archivos cuando realmente se han ejecutado (sync) y todos los usuarios tienen acceso a este directorio con privilegios de usuario anónimo.

También se pueden especificar diferentes opciones para equipos concretos en lugar de la red completa, siguiendo la estructura de la línea de /etc/exports.

Para comprobar que se ha exportado correctamente primero habrá que relanzar el servicio NFS y luego ejecutar la orden:

```
$ sudo exportfs
```

```
/alumnos 192.168.100.0
```

Es muy importante en este archivo /etc/exports tener en cuenta los espacios en blanco que se incluyen. Por ejemplo, las dos líneas siguientes no significan lo mismo:

```
/alumnos 192.168.100.5(rw)
```

```
/alumnos 192.168.100.5 (rw)
```

La primera permite sólo a los usuarios del equipo con IP 192.168.100.5 acceder al directorio

/alumnos en modo lectura y escritura. La segunda permite a los usuarios de 192.168.100.5 montar el directorio /alumnos como de sólo lectura (valor por defecto), pero el resto podría montarlo en modo lectura/escritura.

6.6 Archivos /etc/hosts.allow y /etc/hosts.deny

Los ficheros /etc/hosts.allow y /etc/hosts.deny tienen la siguiente estructura:

```
servicio: host [o red/mascara_subred], host [o red/mascara_subred]
```

Donde:

servicio : servicio permitido o denegado para algunos equipos (IP).
host [o red/mascara_subred] : dirección IP del host de un cliente.

Archivo /etc/hosts.deny
Incluimos todas las restricciones que harán mas seguro nuestro sistema.

En nuestro caso denegamos el acceso a portmap desde cualquier IP. De esta forma sólo tendrán acceso a portmap los equipos que incluyamos en /etc/hosts.allow.

El contenido de /etc/hosts.deny será:

```
portmap:ALL
```

Archivo /etc/hosts.allow
Incluimos a qué equipos permitimos el acceso al servicio de nfs y portmap. Podemos indicar

hosts individuales o una red.

```
portmap:192.168.0.0/255.255.255.0
nfs:192.168.0.0/255.255.255.0
```

Después de configurar estos archivos hay que relanzar los servicios:

```
$ sudo /etc/init.d/nfs-common restart
$ sudo /etc/init.d/nfs-kernel-server restart
```

6.7 Instalación del cliente NFS

Ir a *Sistema > Administrador > Synaptic* y actualizar el sistema pulsando el botón *Recargar*.

Buscar entradas nfs y marcar para instalar el paquete **nfs-common**. Una vez se ha instalado el cliente NFS hay que indicar en el archivo /etc/fstab que se quiere montar el directorio compartido indicando el punto de montaje en el sistema de archivos local.

Para ello abrir el archivo /etc/fstab desde el editor de archivos gedit y añadir una línea para el directorio compartido /alumnos. En esta línea se indica el servidor NFS y nombre del directorio a montar, el punto de montaje local, el tipo de sistema de archivos, y las opciones de montaje.

En nuestro caso indicamos los siguientes valores:

```
192.168.100.1:/alumnos /compartido nfs user,uid=1001,noauto,rsize=4096,wsize=4096,noatime 0 0
```

Indica que, de la máquina 192.168.100.1 (en la que se supone está ejecutándose un servidor NFS) se está compartiendo el directorio /alumnos y que vamos a montar de forma no automática (noauto) en el directorio local /compartido (que deberá existir), este montaje sólo lo podrá realizar el usuario (user) con UID 1001 y se han habilitado bufferes de 4MB para lectura (rsize) y escritura (wsize). La opción noatime no actualiza la fecha de acceso.

Si queremos ahora probar que se monta correctamente el directorio /alumnos en nuestra máquina podemos ejecutar cualquiera de las órdenes siguientes:

```
$ sudo mount -t nfs -o rw 192.168.100.1:/alumnos /compartido
```

```
$ sudo mount -a
```

La primera monta el directorio remoto con las opciones que le pasamos en la orden y la segunda monta todas las entradas del archivo /etc/fstab con las opciones indicadas.

Si ahora queremos comprobar el montaje real del directorio /alumnos, así como su contenido, ejecutamos la orden:

```
$ showmount -e
```

```
Export list for servidor:
```

```
/alumnos 192.168.100.1
```

```
$ ls /compartido
```

```
-rw-r--r-- 1 root root 0 2009-07-30 12:02 proyectoA.txt
```

Como se ha comentado en la Introducción, los dispositivos con sistema de archivos NFTS (Windows) todavía no se pueden exportar y montar utilizando NFS, incluso su versión 4.

7 Opciones de montaje

La tabla siguiente describe otras opciones de montaje disponibles.

Sección
Descripción
intr
Permite la interrupción de las peticiones NFS si el servidor cae o no puede ser accedido por algú
hard/soft
Existen dos tipos de montaje: hard y soft.

Indican si el programa que está utilizando un archivo vía NFS, debe p	***hard*** **parar y esperar (**
Con la opción	***hard***

Con la opción	***soft***

noauto

No montar con la opción -a

noexec

No permite la ejecución de archivos binarios en los sistemas de archivos montados.

noacl

No se utilizan las ACLs.

nohide

Indica que no se 'escondan' otros sistemas de archivos que están montados en algún subdirecto

nolock

Desactiva el bloqueo de archivos.

nosuid

No permite que los bits set-user-identifier o set-group-identifier funcionen. Es muy importante po

nouser

Solo	***root***

sec=sys

La opción sec en general indica el tipo de seguridad que se va a utilizar cuando se autentique ur

sec=sys es la opción por defecto, y utiliza los UIDs y GIDs locales para autenticar las conexiones

sec=krb5

Utiliza Kerberos v5 en vez de los UIDs y GIDs locales para autenticar los usuarios.

sec=krb5p

Utiliza Kerberos v5 para la autenticar los usuarios y encripta el tráfico NFS. Es la configuración r

tcp

Se utiliza el protocolo TCP para el montaje NFS.

udp

Se utiliza el protocolo UDP para el montaje NFS.

Tabla 1: Opciones de montaje de NFS.

8 Instalación y configuración de NFSv4

Como ya se ha comentado, la versión 4 de NFS introduce varias mejoras encaminadas principalmente a la seguridad de NFS. No en vano la 'leyenda urbana' asocia las siglas NFS a No File Secure, debido a los agujeros de seguridad que presenta el servicio.

Características de NFSv4:

-

Autenticación de los usuarios y no de los equipos clientes.

-

Utilización de un único puerto TCP 2049 lo que permite configurar un cortafuegos .

-

No es necesario el demonio portmap.

-

Respecto a seguridad se puede seguir utilizando la seguridad básica del sistema o utilizar Kerberos.

-

La compartición de directorios utiliza un sistema de archivos virtual, similar a los servidores web o servidores FTP.

Vamos a ver una primera aproximación a lo que sería la configuración de un servidor y cliente NFSv4 sin utilizar la autenticación con Kerberos.

8.1 Demonios de NFSv4

NFSv4 además de los demonios ya conocidos nfsd y mountd utiliza otros:

-

rpc.gssd: permite que NFSv4 pueda realizar el proceso de autenticación en la conexión entre cliente y servidor utilizando Kerberos v5. Ejecutado en el cliente NFS.

-

rpc.svcgssd: permite que NFSv4 pueda realizar el proceso de autenticación en la conexión entre cliente y servidor utilizando Kerberos v5. Ejecutado en el servidor NFS.

-

rpc.idmapd: realiza el mapeo entre UIDs/GIDs locales y nombres NFSv4 de usuario (o grupos) del tipo usuario@dominio. Debe ejecutarse tanto en el cliente como en el servidor, ya que la t raducción entre los nombres de usuario y los identificadores de usuario es en ambos sentidos.

Los demonios específicos para Kerberos podemos activarlos y/o desactivarlos desde los archivos de configuración /etc/default/nfs-common y /etc/default/nfs-kernel-server simplemente añadiendo 'no' en las siguientes líneas:

1.

En /etc/default/nfs-common:

```
NEED_GSSD=no
```

2. En /etc/default/nfs-kernel-server:

```
NEED_SVCGSSD=no
```

Si hacemos modificaciones sobre los archivos de configuración hay que reiniciar los servicios:

```
$ sudo /etc/init.d/nfs-common restart
$ sudo /etc/init.d/nfs-kernel-server restart
```

8.2 Servidor NFSv4

En el servidor NFSv4 se necesita:

-

El paquete nfs-kernel-server.

-

Los demonios rpc.nfsd, rpc.mountd y rpc.idmapd funcionando.

-

Únicamente el puerto TCP 2049, si se utiliza la seguridad básica del sistema (sec=sys).

-

Disponer de un sistema de archivos raíz para todos los directorios exportables. Este sistema de archivos será virtual. De esta forma el cliente NFS ya no necesita indicar la ruta física del directorio a montar sino la relativa al sistema de archivos raíz.

-

Configurar los archivos /etc/hosts.allow y /etc/hosts.deny si se quiere tener soporte para las versiones 3 y 4 simultáneamente.

Por ejemplo si el directorio raíz va a ser /exportados, creamos el directorio con los permisos adecuados:

```
$sudo mkdir -m 777 /exportados
```

y añadimos la línea correspondiente en el archivo /etc/exports indicando que es raíz con la opción fsid=0:

```
$sudo gedit /etc/exports
```

```
/exportados 192.168.100.0/24(ro,sync,root_squash,no_subtree_check,fsid=0)
```

Que significa que la red 192.168.100.0/24 tiene acceso en sólo lectura al sistema de archivos raíz de la exportación.

A partir de aquí ya podemos definir otros directorios exportables. Por ejemplo, queremos exportar el directorio /home en modo lectura y escritura. En primer lugar creamos el punto de montaje dentro de /exportados en el propio servidor NFS.

```
$ cd /exporta $ sudo mkdir -m 777 home
```

En el servidor montamos el directorio real en el punto de montaje creado. Editamos el archivo /etc/fstab:

```
/home /exportados/home none rw,bind 0 0
```

La opción de montaje bind indica que lo mismo que tenemos en el directorio /home lo tendremos montado en el directorio /exportados/home.

Debemos forzar el montaje ejecutando la orden:

```
$sudo mount -a
```

Y ahora añadimos en /etc/exports el directorio a exportar referido al sistema de archivos raíz de la exportación. Hay que añadir la opción nohide para que los directorios se vean desde el exterior.

```
/exportados/home 192.168.100.0/24(rw,nohide,sync,root_squash,no_subtree_check)
```

Exportamos:

```
$sudo exportfs -r
```

Comprobamos que se ha exportado correctamente:

```
$showmount -e
```

```
/exportados 192.168.100.0/24


/exportados/home 192.168.100.0/24
```

8.3 Cliente NFSv4

En el cliente el directorio compartido /exportados/home lo vamos a montar en el directorio /compartido, que deberá existir.

Como utilizamos seguridad básica del sistema, en el cliente debe estar ejecutándose sólo el demonio idmapd. En el archivo de configuración /etc/default/nfs-common activamos la opción:

```
NEED_IDMAPD=yes
```

Reiniciamos el cliente NFS:

```
$ sudo /etc/init.d/nfs-common restart
```

Para montar en el cliente NFS el sistema de archivos exportado hay que añadir la siguiente línea en /etc/fstab:

```
192.168.100.1:/ /compartido nfs4 rsize=4096,wsize=4096,hard,intr 0 0
```

Observar como se ha escrito 192.168.100.1:/ y no 192.168.100.1:/exportados ya que en el servidor se ha exportado este sistema de archivos como raíz de todos los sistema de archivos exportados, con fsid=0.

La línea correspondiente al directorio /exportados/home se escribe:

```
192.168.100.1:/home /compartido/home nfs4 rsize=4096,wsize=4096,hard,intr 0 0
```

Montamos todo y comprobamos el montaje:

```
$sudo mount -a
$mount .....
192.168.100.1:/ on /compartido type nfs4 (rw,rsize=4096,wsize=4096,hard,intr,clientaddr=192.168.100.2,addr=192.168.100.2)
192.168.100.1:/home on /compartido/home type nfs4 (rw,rsize=4096,wsize=4096,hard,intr,clientaddr=192.168.100.2,addr=192.168.100.2)
```

Si queremos tener información completa de como se ha realizado el montaje y comprobar las opciones reales de montaje ejecutamos la orden siguiente:

```
$cat /proc/mounts
192.168.100.1:/ /compartido nfs4 rw,vers=4,rsize=4096,wsize=4096,namlen=255,hard,noi
clientaddr=192.168.100.2,addr=192.168.100.1 0 0
192.168.100.1:/home /compartido/home nfs4 rw,vers=4,rsize=4096,wsize=4096,namlen=2
clientaddr=192.168.100.2,addr=192.168.100.1 0 0
```

Comprobamos que el protocolo obligatorio para NFSV4 es TCP (proto=tcp) y se considera que la seguridad es la de sistema (sec=sys).

8.4 Otros detalles de NFSv4

Al ejecutar la orden mount hemos visto que se ha añadido de forma automática una nueva línea que hace referencia a **rpc_pipefs**. En la v4 de NFS es necesario indicar dónde se van a guardar las tuberías o pipes utilizadas en la comunicación entre los procesos. Lo podemos comprobar en el archivo /etc/fstab:

```
rpc_pipefs /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
```

Podemos comprobar que existe el directorio /var/lib/nfs/rpc_pipefs y que está referenciado en el archivo de configuración /etc/idmapd.conf.

Respecto al demonio rpc_idmapd decir que su configuración viene determinada en /etc/idmapd.conf y su función es realizar la traducción entre nombres de usuario y los identificadores de usuario, en ambos sentidos.

9 Seguridad NFS

Si queremos que nuestro servicio NFS sea mas seguro deberíamos tener en cuenta una serie de detalles, como son:

1.

Utilizar los comodines (metacaracteres) lo menos posible, ya que podemos dar acceso a más equipos de los que estamos pensando.

2.

Utilizar reglas de Iptables (cortafuegos) para limitar el acceso a los puertos utilizados por los demonios del servicio NFS.

3.

El uso de los archivos /etc/hosts.allow y /etc/hosts.deny no es obligatorio pero es preferible configurarlos para garantizar la seguridad de los datos.

4.

Exportar sistemas de archivos de lectura (ro) siempre que sea posible.

5.

El dueño de los archivos y directorios exportados que sea root ya que es posible mapear el UID de root al del usuario nobody.

6.

Intentar que los archivos exportados no tengan permiso de escritura para el grupo (ACL).

7.

Las versiones 2 y 3 de NFS no disponen de control de acceso para los usuarios concretos. En ellas, cuando un sistema de archivos es exportado, cualquier usuario en cualquier máquina remota conectada al servidor NFS puede acceder a los datos compartidos. El único mecanismo de seguridad que tienen es utilizar el acceso de sólo lectura y reducir todos los usuarios a uno común cuyo UID y GID especificamos.

8.

Si no se utiliza la opción de exportación *squash*, cualquier usuario root en el equipo cliente puede convertirse en un usuario con acceso privilegiado simplemente ejecutando la orden:
su -
. Conviene siempre tener activada alguna opción de
squash
.

9.

La versión mas segura de NFS es la 4.

10 Conclusión

Hemos visto como el servicio NFS proporciona una solución a la necesidad de compartir archivos y directorios entre sistemas heterogéneos (con la salvedad de NTFS) así como controlar sus permisos de acceso de forma consistente. Con la ventaja de que estos sistemas pueden tener hardware distinto.

NFS crea una capa de abstracción que permite a los usuarios acceder a archivos y/o ejecutar programas ubicados en equipos remotos como si fueran locales. Con todas sus ventajas e inconvenientes, relativos sobre todo a los temas de seguridad.

Con la aparición de la versión 4 se ha dado un gran paso hacia la comunicación segura incorporando mecanismos de autenticación de usuarios así como listas de control de acceso que especifican los permisos concretos para cada recurso compartido.

Pero... siempre hay un pero... No se puede todavía comparar NFS con la potencia de la compartición vía SAMBA, nativa en Windows pero completamente multiplataforma. Las distribuciones GNU/Linux la soportan permitiendo una interoperabilidad entre los sistemas completa, aunque con una complejidad en su configuración importante. Parece que el diálogo Windows-GNU/Linux no acaba de ser todo lo fluido que quisiéramos todos, aunque con SAMBA4 parece que puede venir la solución.

Notas

[1]Como es un archivo de sistema habrá que lanzar gedit con sudo: $sudo gedit /etc/exports.