

T01 - Gestión de servicios (2,25 Puntos)

NFS (Network File System)

Laboratorio de Software de Comunicaciones

Área de Ingeniería Telemática

Antonio Espinosa Guerrero

Alsc10

©LSC 2004/2005

ÍNDICE

ÍNDICE	2
1. Objetivos y Alcance.....	4
1.1. Introducción.....	4
1.2. Motivación y Funcionalidad del Servicio.....	4
1.3. Documentación Bibliográfica.....	5
2. Base Teórica	6
2.1. Descripción del Servicio y Conceptos Implicados	7
2.2. Análisis de Protocolos	8
2.2.1. Funcionamiento y Estructura del Protocolo	10
2.2.2. Uso del Protocolo en el Servicio.....	13
3. Solución Adoptada para Ofrecer el Servicio.....	14
3.1. Soluciones Existentes en el Mercado	15
3.2. Referencias y Características de la Solución Adoptada.....	15
3.3. Equipamiento Necesario	16
4. Parámetros de Configuración del Servidor	16
5. Proceso de Instalación/Administración del Servidor	21
5.1. Obtención del Código.....	22
5.2. Instalación del Servidor	22
5.3. Configuración del Servidor	24
5.4. Puesta en Funcionamiento del Servicio	27
5.5. Administración y Monitorización del Funcionamiento	31
5.6. Pruebas del Servicio	31

6.	Análisis del Intercambio de Mensajes por la Red.....	34
7.	Interfaz Gráfica de Gestión.....	36
8.	Ampliaciones/Mejoras del Servicio	37
9.	Incidencias y Principales Problemas Detectados.....	38
10.	Resumen y Conclusiones	40

1. Objetivos y Alcance

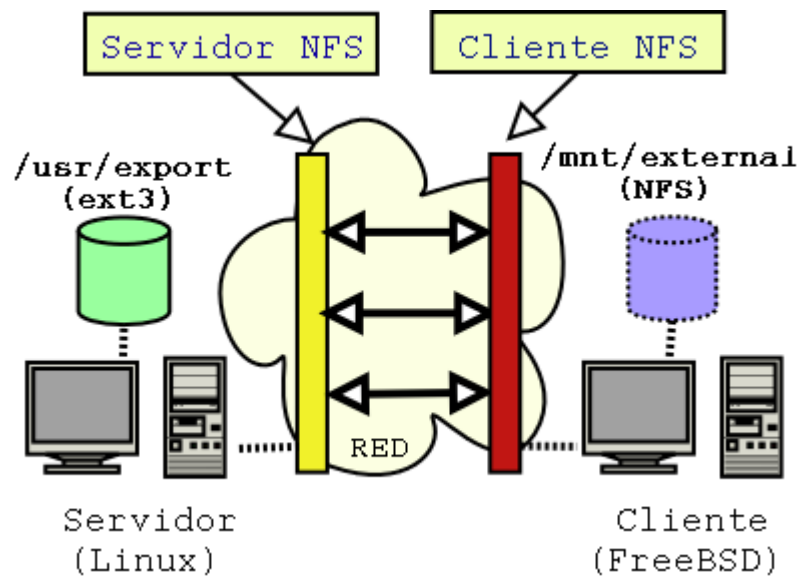
1.1. Introducción

Linux al igual que Unix brinda la posibilidad de compartir recursos entre las máquinas a través del Sistema de Ficheros de Red o NFS (*Network File System*). Este servicio básicamente permite el acceso de un cliente NFS a las partes del sistema de ficheros que sean exportadas por un servidor NFS. El cliente en dependencia de las restricciones que le sean impuestas podrá manipular los ficheros importados como parte de su propio sistema de ficheros. Para configurar este servicio se deben tener en cuenta los problemas de seguridad que puede generar una mala concepción del mismo. No obstante es muy conveniente y fácil de emplear en numerosos contextos donde la seguridad no sea un requisito indispensable.

1.2. Motivación y Funcionalidad del Servicio

NFS no es en realidad un sistema de archivos físico, sino una capa de abstracción del sistema de archivos real (ext2, UFS, FFS, etc...) que permite el montaje de éste remotamente. Por ejemplo, si nuestro servidor es una máquina Linux que exporta por NFS un directorio llamado, pongamos, `/usr/export` cuyo sistema de archivos es ext3, y tenemos un cliente FreeBSD que quiere montar de manera remota ese directorio como `/mnt/external`, no será necesario que nuestro FreeBSD tenga soporte para ext3, sino que simplemente lo tenga para ser cliente NFS.

Para entender este ejemplo mejor vemos un gráfico ilustrativo de la funcionalidad del servicio:



1.3. Documentación Bibliográfica

Los libros que he consultado son:

- *Managing NFS and NIS* por Hal Stern, Mike Eisler, y Ricardo Labiaga; O'Reilly & Associates — Es una guía de referencia excelente para las diferentes opciones NFS disponibles de montaje y exportación.
- *NFS Illustrated* por Brent Callaghan; Addison-Wesley Publishing Company — Proporciona comparaciones entre NFS y otros sistemas de ficheros de red y muestra, en detalle, como las comunicaciones NFS funcionan.
- *Edición Especial LINUX* de Prentice Hall. Muy buen libro sobre Linux en general.

- *Serie Práctica LINUX* por M. Drew Streib, Michael Turner, et al. — Es un libro básico pero muy claro.

En cuanto a webs visitadas podemos extraer algunas de las más importantes:

- <http://nfs.sourceforge.net/>
- <http://www.elrincondelprogramador.com>
- <http://www.linux-es.org>
- <http://www.linuxdata.com.ar>
- <http://www.linux.cu/manual/>
- <http://fferrer.dsic.upv.es/cursos/Linux/basico>
- <http://es.tldp.org>
- www.linuxdoc.org
- <http://www.tldp.org/HOWTO/NFS-HOWTO/>

2. Base Teórica

NFS utiliza al igual que NIS (Network Information Service), RPC (Remote Procedure Calls). En Linux esto es posible, gracias a una mezcla de funcionalidad del Kernel en el cliente y un demonio servidor de NFS en el servidor.

La forma de trabajar de NFS es la siguiente. Un cliente intenta montar (conectar a su árbol de directorios) un directorio desde un host remoto en un directorio local de la misma forma que si fuera un dispositivo físico. Sin embargo la sintaxis empleada para montar el directorio remoto es diferente:

```
[root@mis01]# mount -t nfs hostname:/dirremoto /dirlocal
```

El comando mount intenta conectar con el demonio mountd en el host remoto vía RPC. El servidor chequeará si el cliente tiene permitida la operación, y si es así le devolverá un manejador de fichero.

2.1. Descripción del Servicio y Conceptos Implicados

Cuando alguien accede a un fichero a través de NFS, el kernel coloca una llamada RPC en el demonio nfsd en la máquina servidora. Esta llamada toma el manejador de fichero, el nombre de fichero a ser accedido, y el uid y el gid del usuario como parámetros, que se utilizan para determinar los derechos de acceso.

En la mayoría de implementaciones UNIX, la funcionalidad NFS, tanto en el cliente como en el servidor se implementa como demonios (procesos) a nivel de Kernel que se arrancan desde el espacio de usuario en el boot del sistema. Estos son, el demonio nfsd en el host servidor y el demonio de bloqueo de E/S biod en el cliente.

El NFS de Linux es un poco diferente ya que el código del cliente está integrado en el nivel del Sistema de Ficheros Virtual (VFS) del kernel y no requiere control adicional a través de un demonio biod.

Actualmente existen dos implementaciones del servidor NFS bajo Linux. Una es el servidor NFS en el espacio del usuario y otra el servidor NFS en el espacio del Kernel. La del espacio del usuario tiene que copiar memoria extra entre el espacio del kernel y el espacio del usuario y además es una sobrecarga para el cambio de contexto. No se permiten bloqueos a nivel de registro ni de fichero. Mientras que el servidor NFS en el espacio del kernel no tiene que mover memoria entre los dos espacios ya que se ejecuta en el espacio del kernel y realiza las llamadas RPC dentro del kernel. Si que

permite bloqueo de registros y ficheros lo cual es importante cuando tienes un entorno heterogéneo, y además permite lanzar varias del demonio servidor.

RedHat Linux implementa por defecto el servidor NFS en el espacio del Kernel.

2.2. Análisis de Protocolos

Lo que comúnmente se llama NFS está formado por 4 protocolos distintos. Cada uno depende de las *Remote Procedure Calls* (RPC) y de portmap (también llamado rpc.portmap). Un portmapper convierte números de programa RPC en números de puerto. Cuando un servidor RPC se inicia, dice a portmap qué puerto usará y el número de programa RPC manejado. Cuando un cliente quiere enviar una petición RPC a un número de programa dado, primero contacta con el servidor portmap para tomar el número de puerto dando acceso al programa deseado. Después, dirige los paquetes RPC al puerto correspondiente.

Los 4 servicios que permiten funcionar a NFS son:

Protocolo	Descripción	Demonio
nfs	Este protocolo es el básico y permite crear, buscar, leer o escribir ficheros. Este protocolo también maneja autenticación y estadísticas de ficheros.	nfsd
mountd	Éste se encarga de montar sistemas exportados para acceder a ellos con nfs . El servidor recibe peticiones como mount y umount debiendo mantener información sobre los	mountd

	sistemas de ficheros exportados.	
nsm (Network Status Monitor)	Se usa para monitorizar los nodos de la red y así conocer el estado de una máquina (cliente o servidor). Informa, por ejemplo, de un re arranque.	statd
nlm (Network Lock Manager)	Para impedir modificaciones de los datos por varios clientes al mismo tiempo, este protocolo maneja un sistema de bloqueo. Así, con la ayuda del protocolo Nsm es posible conocer cuándo se está reiniciando un cliente. Nsm libera todos los bloqueos del cliente antes de devolverlos.	lockd

El demonio `knfsd`, disponible con las últimas versiones del núcleo, soporta directamente los protocolos **nfs** y **nlm**. Por otro lado, **mountd** y **nsm** no están todavía soportados. Cuando el servidor NFS está instalado y arrancado, podemos verificar que todo esté funcionando con el comando:

```
>> ps auxwww | egrep "nfs/mount/lock/stat"
root 1370 0.0 0.2 1176 580 ? S 22:28 0:00 rpc.mountd --no-nfs-version
3
root 1379 0.0 0.0 0 0 pts/0 SW 22:28 0:00 [nfsd]
root 1380 0.0 0.0 0 0 pts/0 SW 22:28 0:00 [nfsd]
root 1381 0.0 0.0 0 0 pts/0 SW 22:28 0:00 [nfsd]
root 1382 0.0 0.0 0 0 pts/0 SW 22:28 0:00 [nfsd]
root 1383 0.0 0.0 0 0 pts/0 SW 22:28 0:00 [nfsd]
```

```

root    1384  0.0  0.0    0    0  pts/0    SW   22:28   0:00  [nfsd]
root    1385  0.0  0.0    0    0  pts/0    SW   22:28   0:00  [nfsd]
root    1386  0.0  0.0    0    0  pts/0    SW   22:28   0:00  [nfsd]
root    1399  0.0  0.0    0    0  pts/0    SW   22:28   0:00  [lockd]
root    1409  0.0  0.2  1156  560  ?        S    22:28   0:00  rpc.statd
root    1652  0.0  0.1  1228  484 pts/3    S    22:49   0:00  egrep nfs/mount/lock/stat

```

Por el momento, están disponibles dos versiones de NFS (versiones 2 y 3, que para distinguirlas denotaremos NFSv2 y NFSv3, respectivamente). Los servidores NFS de Linux sólo soportan la versión 2 (de aquí la opción en la línea mountd del ejemplo anterior).

NFS trata con una estructura de datos llamada *file handle*. Es una serie de bits bastante esotérica que permite identificar de forma única cada objeto del sistema de ficheros (como un fichero, pero no tan sólo ficheros). Contiene por ejemplo el ínodo del fichero y también una entrada representando el dispositivo donde se localizan. Por tanto, podemos ver NFS como un sistema de ficheros dentro de otro sistema de ficheros.

2.2.1. Funcionamiento y Estructura del Protocolo

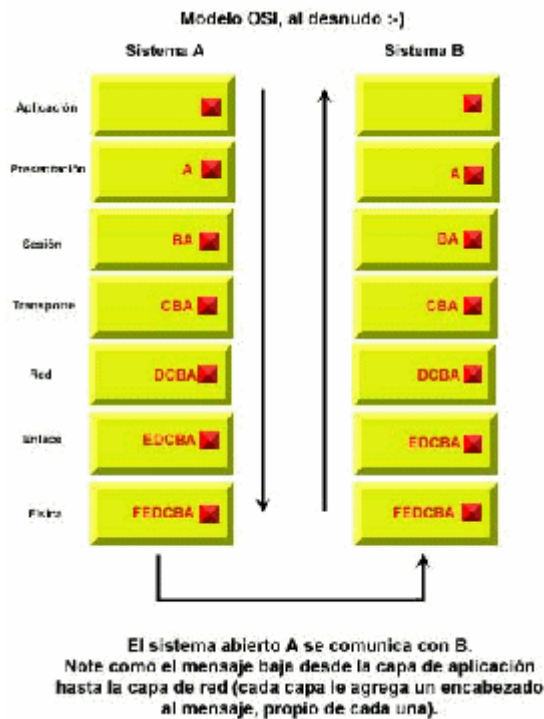
NIS y NFS utilizan protocolos de redes para comunicarse con otras máquinas en la red. Se hará a continuación una pequeña introducción al modelo OSI y TCP/IP [1] [2] y se verá en donde encajan NIS y NFS dentro de ese modelo.

Modelo OSI y TCP/IP

Para poder conectar dos aplicaciones que están separadas físicamente entre sí, se utiliza un lenguaje y reglas de comunicación común a ambas, llamado *protocolo de comunicación*. Existe una abstracción llamada el modelo **OSI**, la cual permite a aplicaciones remotas comunicarse de manera confiable, a la vez de que su

funcionamiento se mantiene independientemente del medio físico por el cual entran en contacto.

El modelo OSI está compuesto de siete capas, cada una de las cuales es recorrida de arriba abajo, las cuales permiten la comunicación entre una aplicación y otra:



A continuación se muestra el significado de cada capa en el modelo OSI y como NIS y NFS encajan en ese modelo:

1. Capa de aplicación. Esta capa trata con los detalles específicos de la aplicación. *NIS* y *NFS* son protocolos que utilizan otras capas inferiores para lograr su cometido.

2. Capa de presentación. La capa de presentación trata con la representación de los datos que los componentes de la capa de aplicación usan o refieren en sus comunicaciones. Sun desarrolló un protocolo llamado *XDR* (*External Data Representation, representación externa de datos*) el cual se encarga de colocar

los datos en forma *canónica* para que estos puedan ser llevados de plataforma a plataforma sin ningún problema.

3. Capa de sesión. La capa de sesión provee mecanismos para organizar y sincronizar intercambios de datos con las capas superiores. Esto se debe a las diferencias en como una arquitectura almacena la información respecto a la otra. En esta capa se alojan las llamadas a procedimientos remotos (*RPC, Remote Procedural Call*). En vez de ejecutar un proceso en la máquina local, el proceso es ejecutado en la máquina remota (típicamente porque el recurso que necesita el proceso no se encuentra en la máquina local). *RPC* por lo general utiliza *UDP* para comunicarse ya que es más rápido que *TCP*.

4. Capa de transporte. Le asegura a las capas superiores que los datos serán transmitidos de manera confiable y con el menor costo posible. *TCP* (Uno de los protocolos sobre los cuales trabaja Internet) se encuentra aquí. *UDP* también acompaña a *TCP* en esta capa. La diferencia básica entre *TCP* y *UDP* es que *TCP* verifica errores en la transmisión mientras que *UDP* no. Una máquina puede tener varias conexiones *TCP* o *UDP* a la vez, por lo que se utiliza *un puerto* para identificar cada conexión.

5. Capa de red. Se encarga de considerar el enrutamiento de la información. *IP* (Uno de los protocolos sobre los cuales trabaja Internet) se encuentra aquí. El protocolo *IP* trabaja con el concepto de direcciones *IP*, las cuales identifican de manera única a una máquina en la red. Una dirección *IP* está formada por un grupo de *4 octetos*, separados por punto (Por ejemplo *150.185.146.1* es una dirección *IP*).

6. Capa de enlace. Provee una transferencia punto a punto. También detecta errores provenientes de la capa física. El protocolo Ethernet se encuentra aquí. Ethernet tiene un grupo de direcciones de *48 bits* llamadas *MAC (Media Access Control)*. Por ejemplo, *8:0:20:ae:6:1f* es una dirección Ethernet.

7. Capa física. Se encarga de los medios electrónicos y mecánicos que comienzan, mantienen y detienen las conexiones físicas entre dos entidades de enlace.

Por ejemplo, la fibra óptica se encuentra en esta capa. Es en esta capa donde se encuentran términos como *MTU* (Maximun Transfer Unit, Unidad Máxima de Transmisión) la cual especifica el tamaño máximo de un paquete que puede ser enviado por la red.

El protocolo TCP/IP no encaja exactamente dentro del modelo de referencia OSI, pero lo sigue de cerca.

2.2.2. Uso del Protocolo en el Servicio

NIS y *NFS* trabajan utilizando el modelo cliente - servidor. Básicamente, un cliente es una entidad que solicita un servicio y un servidor es la entidad que provee el recurso solicitado por el cliente.

En este esquema, RPC juega un papel importante ya que *NIS* y *NFS* se basan en sus servicios.

En vez de ejecutar el procedimiento en una máquina local, RPC pasa una serie de argumentos al procedimiento en un datagrama de red. El cliente RPC crea la sesión al localizar al servidor apropiado y se envía el datagrama a un proceso en el servidor que pueda ejecutar la llamada del procedimiento remoto. En el servidor el argumento es desempquetado, el servidor ejecuta el comando y retorna las respuestas, si hay alguna, al cliente. De vuelta en el cliente, el valor del RPC es convertido a un valor esperado por la función que lo llamo y la aplicación continua como si un procedimiento local hubiera sido llamado.

La ubicación de puertos para una sesión es manejada por un demonio llamado *portmap*.

3. Solución Adoptada para Ofrecer el Servicio

NFS se apoya en las llamadas de procedimientos remotos (RPC) para funcionar. Se requiere `portmap` para trazar las peticiones RPC a los servicios correctos. Los procesos RPC notifican a `portmap` cuando comienzan, revelando el número de puerto que ellos están monitorizando y el número de programas RPC que esperan servir. El sistema cliente entonces contacta con el `portmap` del servidor con un número de programa RPC particular. Entonces `portmap` redirecciona al cliente al número del puerto apropiado para que se comunique con el servicio apropiado.

Como los servicios basados en RPC confían en `portmap` para hacer todas las conexiones con las peticiones de clientes entrantes, `portmap` debe estar disponible antes que cualquiera de esos servicios comience. Si, por alguna razón, el servicio `portmap` inesperadamente se quita, reinicie `portmap` y cualquier servicio que estuviera ejecutándose entonces.

El servicio `portmap` puede ser usado con los ficheros de accesos de máquinas (`/etc/hosts.allow` y `/etc/hosts.deny`) para controlar a qué sistemas remotos les son permitidos usar servicios basados en RPC en su máquina.

Los privilegios de montajes NFS son permitidos específicamente a máquinas, no a usuarios. Si permite a un sistema acceder a una parte en concreto de su disco duro, los usuarios de esa máquina podrán acceder a esos datos compartidos.

Al configurar el fichero `/etc/exports`, sea extremadamente cuidadoso cuando comparta directorios con permisos de lectura y escritura (`rw`) a un sistema remoto. Los usuarios de sistemas remotos que monten su sistema de ficheros exportados, pueden modificar los datos.

3.1. Soluciones Existentes en el Mercado

NFS es un protocolo que data de los años 80. En esas fechas los problemas de seguridad eran menores. Todavía podían construirse protocolos basados en la confianza, tanto el servidor como el cliente confiando en la información que intercambian. Internet convierte este principio en algo absurdo y este es sin duda uno de los mayores problemas de NFS. La versión 2 del protocolo es la primera versión publicada y sigue la siendo la mas extendida en nuestros días.

Existen implementaciones sobre varias plataformas diferentes y se han descrito pocos problemas de compatibilidad.

La versión 3 del protocolo data de 1992 y presenta varias mejoras: Mejora del rendimiento debido a la reescritura del código de la red, y al uso de paquetes de datos mayores.

Mejora en la seguridad: Listas de ACL (Listas de control de acceso) que permiten definir acceso a los recursos por UID y fichero a fichero. Implementación de un sistema de autenticación basado en contraseña.

En la actualidad existe una versión 4 pero aún está en desarrollo y por lo tanto no está estandarizada, esta nueva versión pretende mejorar problemas de compatibilidad y de seguridad, existentes en las versiones anteriores.

3.2. Referencias y Características de la Solución Adoptada

La versión 2 de NFS usa el *User Datagram Protocol (UDP)* para proporcionar una conexión de red sin estado entre el cliente y el servidor (la versión 3 de NFS puede usar UDP o TCP corriendo sobre una IP). La conexión UDP sin estado minimiza el tráfico de red, como el servidor NFS manda al cliente una cookie, después el cliente es autorizado a acceder al volumen compartido. Esta cookie, o valor aleatorio que es guardado en la parte del servidor, es pasado por cualquier petición RPC del cliente al servidor. El servidor NFS puede ser reiniciado sin afectar a los clientes y las cookies permanecen intactas.

3.3. Equipamiento Necesario

A nivel de Software necesitamos el equipamiento siguiente:

Kernel: Para poner a funcionar NFS sobre nuestro sistema necesitaremos en primer lugar un kernel con soporte NFS. La mayoría de los kernels que se instalan la primera vez, al ser modulares incluirán el modulo NFS correspondiente y lo cargarán en memoria cuando haga falta. Si no es este el caso, podremos recompilarlo habilitando el soporte de sistema de archivos NFS, y si además queremos que nuestro host sea servidor NFS habilitaremos el soporte correspondiente en el kernel. Aquí cabe hacer algunas salvedades en cuanto a la versión de kernel que utilicen, y el tipo de servicio NFS que pretendan poner en funcionamiento, pero en general sobre núcleos 2.x o superiores, podremos cubrir las necesidades más comunes. Para más detalles, refiéranse a la documentación de NFS.

Aplicaciones: También necesitaremos instalar portmap, y nfs-utils, que contiene todo lo necesario para poner en funcionamiento y administrar el sistema de archivos compartidos

4. Parámetros de Configuración del Servidor

Es sencillo configurar un sistema para compartir ficheros y directorios usando NFS. Cada Sistema de ficheros que se exporta a usuarios remotos vía NFS, así como los derechos de acceso relativos a ellos, es localizado en el fichero `/etc/exports`. Este fichero es leído por el comando `exportfs` que da a `rpc.mountd` y `rpc.nfsd` la información necesaria para permitir el montaje remoto de un sistema de ficheros por una máquina autorizada.

El comando `exportfs` permite exportar o no directorios concretos sin reiniciar los servicios NFS. Cuando se le pasan las opciones apropiadas a `exportfs`, el sistema de ficheros a exportar es incluido en `/var/lib/nfs/xtab`. Como `rpc.mountd` se

refiere al fichero `xtab` para decidir privilegios de acceso a un sistema de ficheros, los cambios en la lista de sistemas de ficheros exportados toman efecto inmediatamente.

Hay varias opciones disponibles cuando usamos `exportfs`:

- `-r` Provoca que todos los directorios listados en `/etc/exports` sean exportados construyendo una nueva lista de exportación en `/etc/lib/nfs/xtab`. Esta opción refresca la lista de exportación con cualquier cambio que hubiéramos realizado en `/etc/exports`.
- `-a` Provoca que todos los directorios sean exportados o no, dependiendo de qué otras opciones hemos pasado a `exportfs`.
- `-o opciones` Permite al usuario especificar directorios a exportar que no estén listados en `/etc/exports`. Estos sistemas de ficheros adicionales compartidos deben ser escritos de la misma forma que son especificados en `/etc/exports`. Esta opción es usada para probar un sistema de ficheros antes de añadirlo permanentemente a la lista de sistemas a exportar.
- `-i` Le dice a `exportfs` que ignore `/etc/exports`; sólo las opciones dadas en la línea de comandos serán usadas para definir los sistemas de ficheros exportados.
- `-u` Termina de exportar directorios que puedan ser montados por usuarios remotos. El comando `exportfs -ua` suspende la compartición de ficheros NFS mientras que mantiene los demonios activos. Para continuar con la compartición NFS, teclee `exportfs -r`.
- `-v` Operación descriptiva, donde los sistemas de ficheros exportados o dejados de exportar son mostrados en gran detalle al ejecutarse el comando `exportfs`.

Si no se pasan opciones al comando `exportfs`, mostrará una lista de los sistemas de ficheros actualmente exportados.

Los cambios efectuados a `/etc/exports` pueden ser leídos al recargar el servicio NFS con el comando `service nfs reload`. Esto deja a los demonios NFS ejecutándose mientras reexporta el fichero `/etc/exports`.

`/etc/exports`

El fichero `/etc/exports` es el estándar para controlar que sistemas de ficheros son exportados a que máquinas, así como para especificar opciones particulares que controlen todo. Las líneas en blanco son ignoradas, se pueden comentar líneas con `#`, y las líneas largas pueden ser divididas con una barra invertida (`\`). Cada sistema de ficheros exportado debe tener su propia línea. La lista de máquinas autorizadas colocada después de un sistema de ficheros exportado, debe estar separada por un espacio. Las opciones para cada uno de las máquinas deben ser colocadas entre paréntesis directamente detrás del identificador de la máquina, sin ningún espacio de separación entre la máquina y el primer paréntesis.

De esta sencilla manera, `/etc/exports` sólo necesita saber el directorio a exportar y las máquinas que pueden utilizarlo:

```
/un/directorio bob.domain.com
/otro/directorio/exportado 192.168.0.3
```

Después de reexportar `/etc/exports` con el comando `/sbin/service nfs reload`, la máquina `bob.domain.com` será capaz de montar `/un/directorio`, y `192.168.0.3` podrá montar `/otro/directorio/exportado`. Como no hay opciones especificadas en este ejemplo, varias preferencias por defecto toman efecto:

- `ro` Sólo lectura (read-only). Las máquinas que monten este sistema de ficheros no podrán cambiarlo. Para permitirles que puedan hacer cambios en el sistema de ficheros, debe especificar la opción `rw` (lectura-escritura - read-write).
- `async` Permite al servidor escribir los datos en el disco cuando lo crea conveniente. Mientras que esto no tiene importancia en un sistema de sólo lectura, si una máquina hace cambios en un sistema de ficheros de lectura-

escritura y el servidor se cae o se apaga, se pueden perder datos. Especificando la opción `sync`, todas las escrituras en el disco deben hacerse antes de devolver el control al cliente. Esto bajará el rendimiento.

- `wdelay` Provoca que el servidor NFS retrase el escribir a disco si sospecha que otra petición de escritura es inminente. Esto puede mejorar el rendimiento reduciendo las veces que se debe acceder al disco por comandos de escritura separados. Use `no_wdelay` para desactivar esta opción, la cual sólo funciona si está usando la opción `sync`.
- `root_squash` Hace que cualquier cliente que acceda al sistema de ficheros exportado (como `root` en la máquina cliente), se convierta en el ID del usuario `nobody`. Esto reconvierte el poder del usuario `root` remoto al de usuario local más bajo, previniendo que los usuarios `root` remotos puedan convertirse en usuarios `root` en el sistema local. Alternativamente, la opción `no_root_squash` lo desactiva. Para reconvertir a todos los usuarios, incluyendo a `root`, use la opción `all_squash`. Para especificar los ID de usuario y grupo para usar con usuarios remotos desde una máquina particular, use las opciones `anonuid` y `anongid` respectivamente. De esta manera, puede crear una cuenta de usuario especial para usuarios NFS remotos para compartir y especificar (`anonuid=<uid-value>,anongid=<gid-value>`), donde `<uid-value>` es el número ID de usuario y `<gid-value>` es el número ID de grupo.

Para saltarse estas opciones predeterminadas, debe especificar la opción que desea cambiar. Por ejemplo, si no especifica la opción `rw`, entonces se exportará en sólo lectura. Cada opción predeterminada debe ser explícitamente sobrescrita con su opción correspondiente. Adicionalmente, hay otras opciones que están disponibles que no afectan a las predeterminadas. Estas incluyen desactivar el navegar por subdirectorios, permitir el acceso a puertos inseguros, y permitir bloquear ficheros

inseguros (necesario para algunas implementaciones antiguas de clientes NFS). Vea la página man de `exports` para estas opciones menos usadas.

Para especificar máquinas a las que permitir usar un sistema de ficheros en concreto, podemos usar varios métodos, entre los que se incluyen:

- *una sola máquina* — Cuando una máquina en particular es especificada con nombre completo de dominio, nombre de máquina o dirección IP.
- *comodines* — Cuando usamos un carácter `*` o `?` para referirnos a un grupo de nombres completos de dominio o direcciones IP o que coincidan con una cadena particular de letras.

Sin embargo, sea cuidadoso cuando use comodines con nombres de dominios completos, e intente ser lo más exacto que pueda. Por ejemplo, el uso de `*.domain.com` como comodín, permitirá a `ventas.domain.com` acceder al sistema de ficheros exportado, pero no a `bob.ventas.domain.com`. Para permitir ambas posibilidades, así como a `sam.corp.domain.com`, debería usar `*.domain.com *.*.domain.com`.

- *redes IP* — Permite el acceso a máquinas basadas en sus direcciones IP dentro de una red más grande. Por ejemplo, `192.168.0.0/15` permite al acceso a las primeras 16 direcciones IP, desde la `192.168.0.0` a la `192.168.0.15`, accedes al sistema de ficheros, pero no a la `192.168.0.16` y superiores.
- *grupos de redes* — Permite que un nombre de grupo de red NIS, escrita como `@<group-name>`, sea usada. Esto pone al servidor NIS controlando el acceso de este sistema de ficheros, donde los usuarios pueden ser añadidos o borrados de un grupo NIS sin que afecte a `/etc/exports`.

Según lo visto en este apartado debemos tener cuidado pues:

La manera en que el fichero `/etc/exports` está organizado es muy importante, particularmente lo que concierne a los espacios en blanco. Recuerde separar siempre

los sistemas de ficheros exportados de máquina a máquina y de uno a otro con un espacio. Sin embargo, no debería haber otros espacios en el fichero a menos que se usen en líneas comentadas.

Por ejemplo, las siguientes dos líneas significan cosas distintas:

```
/home bob.domain.com(rw)
/home bob.domain.com (rw)
```

La primera línea permite sólo a los usuarios de bob.domain.com acceder en modo de lectura-escritura al directorio /home. La segunda línea permite a los usuarios de bob.domain.com montar el directorio de sólo lectura (el predeterminado), pero el resto del mundo puede instalarlo como lectura-escritura. Sea cuidadoso donde se use espacios en blanco en /etc/exports.

5. Proceso de Instalación/Administración del Servidor

Para instalar un servicio NFS sólo necesitamos instalar los paquetes de utilidades más comunes, que son:

```
nfs-utils-clients
nfs-utils
```

Las herramientas del cliente: showmount

Estrictamente hablando solo es necesario el programa mount para hacer funcionar un cliente de NFS. Pero las utilidades de cliente son a menudo muy útiles. Showmount en este caso, nos permite ver la lista de particiones NFS montadas por otras maquinas dentro de la red.

Las herramientas del servidor: mountd, nfsd

Si estamos usando el NFS incluido en el kernel, el trabajo lo lleva a cabo directamente el modulo nfsd.o de Linux. El programa rpc.nfsd solo sirve para

comunicar el portmapper con el kernel. El programa `rpc.mountd` es el programa responsable de la seguridad de los montajes con NFS. Cuando una maquina cliente solicita la exportación de una partición, `mountd` verifica si dicha maquina cliente está autorizada.

5.1. Obtención del Código

El código que debemos de obtener para ejecutar nuestro servicio `nfs`, se limita al paquete `nfs-utils`, este paquete lo podemos descargar de la página: <http://sourceforge.net/projects/nfs/> además no tendremos problemas de compatibilidad de sistemas operativos ya que es un mismo paquete independiente de la plataforma usada.

5.2. Instalación del Servidor

Para instalar un servicio NFS sólo necesitamos instalar los paquetes de utilidades más comunes, que son:

`nfs-utils-clients` (no es obligatorio)

`nfs-utils`

Una vez descargado el paquete correspondiente solo necesitamos descomprimir el paquete, si estaba comprimido, y ejecutar las siguientes instrucciones:

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

Sin embargo, debemos tener mucho cuidado con los problemas derivados de una mala configuración de otros servicios como el cortafuegos.

Uno de los mayores problemas con NFS viene del hecho de que exista por defecto una relación de confianza entre un cliente y un servidor NFS. En el caso de que la cuenta `root` del servidor esté comprometida, la del cliente también lo estará. Un conjunto de medidas esenciales que debe tomarse para conseguir cierta seguridad.

Un cliente no debe confiar ciegamente en un servidor, por ello debemos especificar opciones restrictivas cuando usamos el comando `mount`. Ya hemos mencionado la primera de ellas: `nosuid`. Cancela el efecto de los bits SUID y GID. Con esto alguien que esté como `root` en el servidor primero debe hacer login en el cliente como un usuario normal y después hacerse `root`. Otra opción, más restrictiva, es `noexec`. Prohíbe ejecutar programas en sistema de ficheros exportado. Esta opción únicamente se utiliza en sistemas que sólo contengan datos.

En el lado del servidor NFS, podemos especificar que no confíe en la cuenta `root` del cliente. Tenemos que especificarlo en `/etc/exports` con la opción `root_squash`. Entonces si un usuario con UID 0 (`root`) en el cliente accediese al sistema de ficheros exportado por el servidor, tomaría el UID `nobody` para consultar ficheros. Esta opción está activada por defecto bajo Linux pero se puede desactivar con la opción `no_root_squash`. Se puede especificar qué opciones deben aplicarse en un conjunto de UIDs. Recuerde también que las opciones `anonuid` y `anongid` permiten cambiar los UID/GID de `nobody` por los de otro usuario diferente.

Algunas opciones son más generales y se efectúan por el `portmapper`. Por ejemplo, prohibimos el acceso a todas las máquinas con la siguiente línea en el fichero `/etc/hosts.deny`:

```
# hosts.deny : absolute prohibition for every one to
# use the portmap

portmap: ALL
```

Después en el fichero `/etc/hosts.allow` esta estricta prohibición se puede contrarrestar permitiendo el acceso a las máquinas deseadas.

Unas buenas reglas de cortafuegos también contribuyen a una protección mejor. Observe los puertos usados por los diferentes servicios y los protocolos utilizados:

Servicio RPC	Puerto	Protocolos
portmap	111	udp / tcp
nfsd	2049	udp
mountd	variable	udp / tcp

5.3. Configuración del Servidor

La primera cosa a hacer, como ya hemos visto, es iniciar `portmap` ya que este protocolo es necesario para NFS.

```
root >>/usr/sbin/rpcinfo -p
```

```
rpcinfo: can't contact portmapper: RPC: Remote system  
error - Connection refused
```

```
root >>/sbin/portmap
```

```
root >>/usr/sbin/rpcinfo -p
```

```
program vers proto  port  
100000    2   tcp    111  portmapper  
100000    2   udp    111  portmapper
```

El comando `rpcinfo` muestra los servicios RPCs en la máquina especificada como argumento (opción `-p`). Notamos que `portmap` todavía no está funcionando: lo iniciamos (la mayoría de las distribuciones Linux proveen scripts para automatizar esto en el arranque) y comprobamos que funciona. Otra razón común para que `rpcinfo` responda negativamente es que el `portmapper` no permita la respuesta a

causa de la restricción de seguridad en los ficheros `/etc/hosts.{allow,deny}`. En este caso, añade una entrada "portmap: hosts" en el fichero `hosts.allow`.

Antes de que NFS se inicie por sí mismo, debe ser configurado. Existe un único fichero de configuración que se llama `/etc/exports`. Cada línea muestra la ruta exportada seguida de una lista de clientes a los que se permite el acceso. Se pueden añadir opciones al final de cada nombre de cliente. La página de manual `exports` (`man exports`) explica la sintaxis para los nombres de cliente y las opciones.

Se aceptan como nombres de cliente:

- nombre de la máquina
- caracteres comodín en un nombre de dominio (v.gr. : `linux-*.esi.es`)
- un *netgroup* (`@grupo`) si se usa NIS
- una dirección IP...

No vamos a detallar aquí todas las opciones de montaje disponibles, pero algunas de las más importantes son:

- `rw` (lectura/escritura) : el cliente puede leer y escribir en el sistema exportado
- `ro` (sólo lectura) : el cliente sólo puede leer el sistema exportado
- `root_squash` : es preferible que un usuario *root* del cliente no pueda escribir con permisos de *root*. Para impedirlo, UID/GID 0 (i.e. *root*) en el lado del cliente se traduce en el usuario *nobody*. Esta opción está activada por defecto, pero se puede cancelar con `no_root_squash`
- `all_squash` : todos los clientes que acceden al sistema exportado utilizan el UID/GID de *nobody*

- `anonuid`, `anongid`: el usuario *nobody* ahora usa los UID y GID definidos por estas opciones.

Ahora tenemos que iniciar los demonios `rpc.mountd` y `rpc.nfs` para tener funcionando el servidor NFS. Comprobamos nuevamente que todo está funcionando con el comando `rpcinfo`. Incluso podemos inicializar el servidor para los protocolos **nsm** y **nlm** (`rpc.statd` y `rpc.lockd`, respectivamente). No hay ninguna premisa para arrancar un servidor NFS... pero es altamente recomendable que se reinicie por sí mismo, en caso de que la máquina falle, etc...

Cuando modificamos el fichero de configuración `/etc/exports`, debemos avisar a los demonios implicados que se deben hacer los cambios. El comando `exportfs` transmite esta información a nuestros servidores. La opción `-r` sincroniza el fichero `/etc/mtab` con el fichero `/etc/exports` file. La opción `-v` muestra juntos todos los sistemas de ficheros exportados junto con sus opciones.

Después de ponerse en marcha el servidor NFS, los siguientes ficheros contienen información importante:

- `/var/lib/nfs/rmtab` : cada línea muestra el nombre del cliente y el sistema de ficheros importado desde este servidor;
- `/var/lib/nfs/etab`: el fichero `/etc/exports` sólo contiene una lista de peticiones. `etab` está creado por `exportfs`. Contiene en cada línea información detallada sobre las opciones usadas cuando se exporta un sistema de ficheros a un solo cliente. Es el fichero de referencia usado por `rpc.mountd` cuando es arrancado
- `/proc/fs/nfs/exports` contiene la lista de clientes conocida por el núcleo

- `/var/lib/nfs/xtab`: Se usa por precisión cuando `etab` contiene nombres de clientes y grupos de máquinas con comodines. Este fichero sólo contiene nombres explícitos de máquinas.

Cuando un cliente quiere acceder a un sistema de ficheros, empieza haciendo una petición `mountd`. Entonces se busca en `etab` si la petición está disponible. Se comprueba el núcleo para saber si el cliente tiene permitida la petición (comprobando `hosts.{allow, deny}`, reglas de cortafuegos, ...). El núcleo utiliza `exportfs` para la comprobación, permitiendo actualizar el fichero `/var/lib/nfs/etab`. Si, en este fichero, el sistema exportado tiene permitido ser exportado al grupo al que pertenece el cliente, entonces `mountd` informa al núcleo que actualice `xtab` con este nuevo host.

5.4. Puesta en Funcionamiento del Servicio

Para poner en funcionamiento nuestro servicio NFS, una vez configurado el servidor, sólo necesitamos montar la partición compartida por el servidor en el cliente, teniendo para ello varias opciones.

Cualquier compartición NFS puesta a disposición por un servidor puede ser montada usando varios métodos. Desde luego que puede ser montada manualmente usando el comando `mount`, para adquirir el sistema de ficheros exportado como un punto de montaje concreto. Sin embargo, esto requiere que el usuario `root` teclee el comando `mount` cada vez que el sistema reinicie. Además, el usuario `root` debe recordar desmontar el sistema de ficheros cuando apague la máquina. Otros métodos de configurar los montajes NFS incluyen el modificar `/etc/fstab` o utilizar el servicio `autofs`.

/etc/fstab

Colocando una línea adecuadamente formada en el fichero */etc/fstab* tiene el mismo efecto que el montaje manual del sistema de ficheros exportado. El fichero */etc/fstab* es leído por el script */etc/rc.d/init.d/netfs* cuando arranca el sistema. Los sistemas de ficheros montados, incluyendo NFS, son puestos en su sitio.

Un ejemplo de línea */etc/fstab* para montar un NFS exportado será parecida a:

```
<server-host>:</path/to/shared/directory>  
</local/mount/point> nfs <options> 0 0
```

La opción *<server-host>* tiene que ver con el nombre de la máquina, dirección IP o nombre de domain totalmente cualificado del servidor que exporta el sistema de ficheros. *</path/to/shared/directory>* le dice al servidor que exporta para montar. *</local/mount/point>* especifica dónde, en el sistema de ficheros local, monta el directorio exportado. Este punto de montaje debe existir antes en */etc/fstab* de donde es leído o el montaje fallará. La opción *nfs* especifica el tipo de sistema de ficheros montado.

El área *<options>* especifica como el sistema de ficheros es montado. Por ejemplo, si las opciones *rw,suid* en un montaje en concreto, el sistema de ficheros exportado será montado en modo de lectura-escritura y los ID de usuario y grupo puestos por el servidor serán usados. Aquí no se usan paréntesis.

autofs

Una desventaja de usar */etc/fstab* es que, sin tener en cuenta cuanto use este sistema de ficheros montado, su sistema debe dedicar recursos a guardar este montaje en su sitio. Esto no es un problema con uno o dos montajes, pero cuando su sistema está manteniendo montajes en una docena de sistemas al tiempo, el rendimiento

global puede decaer. Una alternativa a `/etc/fstab` es usar la utilidad basada en el kernel `automount`, la cual monta y desmonta sistemas de ficheros NFS automáticamente, salvando recursos.

El script `autofs`, localizado en `/etc/rc.d/init.d`, es usado para controlar a `automount` a través del fichero de configuración primario `/etc/auto.master`. Mientras que `automount` puede ser especificado en la línea de comandos, es más conveniente especificar los puntos de montaje, nombres de máquinas, directorios exportados y opciones en un conjunto de ficheros que teclearlo todo a mano. Ejecutando `autofs` como servicio que empieza y termina en sus niveles de ejecución designados, las configuraciones de montaje de varios ficheros pueden ser implementados automáticamente. Para usar `autofs`, debe tener al paquete RPM `autofs` instalado en su sistema.

Los ficheros de configuración `autofs` están fijados en una relación padre-hijo. Un fichero principal de configuración (`/etc/auto.master`) se refiere a los puntos de montaje de su sistema que están enlazados a un particular *tipo de mapa*, el cual toma la forma de otros ficheros de configuración, programas, mapas NIS y otros métodos de montaje menos comunes. El fichero `auto.master` contiene líneas referidas a cada punto de montaje, organizadas como:

```
<mount-point>    <map-type>
```

La opción `<mount-point>` indica dónde el dispositivo o sistema de ficheros exportado debe montarse en su sistema local. `<map-type>` muestra la manera como el punto de montaje debe ser montado. El método más común de automontaje de exportaciones NFS es usar un fichero como tipo de mapa del punto de montaje particular. el fichero mapa, usualmente llamado `auto.<mount-point>`, donde `<mount-point>` es el punto de montaje designado en `auto.master`, contiene líneas parecidas a:

```
<directory> <mount-options>  
<host>:<exported-filesystem>
```

<directory> se refiere al directorio dentro del punto de montaje donde el sistema de ficheros exportado debe ser montado. A menudo, como en el comando estándar `mount`, la máquina que exporta el sistema de ficheros y el sistema de ficheros que está siendo exportado, son requeridos en la sección *<host>:<exported-filesystem>*. Para especificar las opciones particulares para montar un sistema de ficheros exportado, colóquelas en la sección *<mount-options>*, separadas por comas. Para montajes NFS que usen `autofs`, debería colocar definitivamente `-fstype=nfs` en la sección *<mount-options>*, como mínimo.

Cómo los ficheros de configuración de `autofs` pueden ser usados para una gran variedad de montajes de muchos tipos de dispositivos y sistemas de ficheros, son particularmente útiles para crear montajes NFS. Por ejemplo, algunas organizaciones guardan un directorio `/home` de usuario en un servidor central via compartición NFS. Entonces, configuran el archivo `auto.master` en cada una de las estaciones de trabajo para que apunten a un fichero `auto.home` que contiene como montar el directorio `/home` via NFS. Esto permite al usuario acceder a sus datos personales y ficheros de configuración en su directorio `/home` conectándose desde cualquier sitio de la red interna. El fichero `auto.master` en esta situación debería parecerse a:

```
/home    /etc/auto.home
```

Esto hace que el punto de montaje `/home` del sistema local sea configurado con el fichero `/etc/auto.home`, que debe ser similar a:

```
*        -fstype=nfs,soft,intr,rsize=8192,wsiz=8192,nosuid  
server.domain.com:/home/&
```

Esta línea establece que cualquier directorio bajo el directorio local /home al que un usuario intente acceder (debido al asterisco), debe resultar en un punto de montaje NFS en el sistema server.domain.com dentro del sistema de ficheros exportado /home. Las opciones de montaje especifican que cada directorio montado via NFS /home debe usar una particular colección de opciones.

5.5. *Administración y Monitorización del Funcionamiento*

NFS no dispone de un archivo log específico en el que “apuntar” cada concurrencia del servicio, por ello usa un log general situado en: /var/log/message. Para ver la monitorización del servicio vamos a describir los errores más comunes que aparecen en este fichero:

- Sale cuando el cliente no tiene permiso de escritura sobre la partición montada.

```
Jan 7 09:15:29 server kernel: fh_verify: mail/guest permission failure, acc=4, error=13
Jan 7 09:23:51 server kernel: fh_verify: ekonomi/test permission failure, acc=4, error=13
```

- Cuando el cliente no obtiene respuesta del cliente, puede ser por la congestión de la red.

```
kernel: nfs: server server.domain.name not responding, still trying
kernel: nfs: task 10754 can't get a request slot
kernel: nfs: server server.domain.name OK
```

- Tras montar el directorio puede salir el siguiente mensaje, indicando que debemos de actualizar la version de mount o de nfs-utils.

```
nfs warning: mount version older than kernel
```

5.6. *Pruebas del Servicio*

Como prueba vamos a ver el caso particular de un servidor que desea compartir una carpeta con un cliente dentro de una misma red de área local.

Procederemos a determinar que directorio se va a compartir. Puede crear también uno nuevo:

```
mkdir -p /var/nfs/publico
```

Una vez hecho esto, necesitaremos establecer que directorios en el sistema serán compartidos con el resto de las máquinas de la red, o bien a que máquinas, de acuerdo al DNS o `/etc/hosts` se permitirá el acceso. Esto deberemos agregarlos en `/etc/exports` determinado con que máquinas y en que modo lo haremos. Se puede especificar una dirección IP o bien nombre de alguna máquina, o bien un patrón común con comodín para definir que máquinas pueden acceder. De tal modo podemos utilizar el siguiente ejemplo (la separación de espacios se hace con un tabulador):

```
/var/nfs/publico *.mi-red-local.org(ro, sync)
```

En el ejemplo anterior se está definiendo que se compartirá `/var/nfs/publico/` a todas las máquinas cuyo nombre, de acuerdo al DNS o `/etc/hosts`, tiene como patrón común **mi-red-local.org**, en modo de lectura escritura. Se utilizó un asterisco (*) como comodín, seguido de un punto y el nombre del dominio. Esto permitirá que *como_se_llame.mi-red-local.org*, *como_sea.mi-red-local.org*, *lo_que_sea.mi-red-local.org*, etc., podrán acceder al volumen `/var/nfs/publico/` en modo solo lectura. Si queremos que el acceso a este directorio sea en modo de lectura y escritura, cambiamos (ro) por (rw):

```
/var/nfs/publico *.mi-red-local.org(rw, sync)
```

Ya que se definieron los volúmenes a compartir, solo resta iniciar o reiniciar el servicio `nfs`. Utilice cualquiera de las dos líneas dependiendo el caso:

```
/sbin/service nfs start  
/sbin/service nfs restart
```

A fin de asegurarnos de que el servicio de `nfs` esté habilitado la siguiente vez que se encienda el equipo, debemos ejecutar lo siguiente:

```
/sbin/chkconfig --level 345 nfs on
```

Este comando hace que se habilite `nfs` en los niveles de ejecución 3, 4 y 5.

Como medida de seguridad adicional, si tiene un contrafuegos o *firewall* implementado, cierre, para todo aquello que no sea parte de su red local, los puertos tcp y udp 2049, ya que estos son utilizados por NFS para escuchar peticiones.

Configurando las máquinas clientes.

Para probar la configuración, es necesario que las máquinas clientes se encuentren definidas en el DNS o en el fichero */etc/hosts* del servidor. Si no hay un DNS configurado en la red, deberán definirse los nombres y direcciones IP correspondientes en el fichero */etc/hosts* de todas las máquinas que integran la red local.

Como root, en el equipo cliente, ejecute el siguiente comando para consultar los volúmenes exportados (-e) a través de NFS por un servidor en particular:

```
showmount -e 192.168.1.254
```

Lo anterior mostrará una lista con los nombres y rutas exactas a utilizar.

Ejemplo:

```
Export list for 192.168.1.254:
/var/nfs/publico          192.168.1.0/24
```

A continuación creamos, como *root*, desde cualquier otra máquina de la red local un punto de montaje:

```
mkdir /mnt/servidornfs
```

Y para proceder a montar el volumen remoto, utilizaremos la siguiente línea de comando:

```
mount servidor.mi-red-local.org:/var/nfs/publico
/mnt/servidornfs
```

Si por alguna razón en el DNS de la red local, o el fichero */etc/hosts* de la máquina cliente, decidió no asociar el nombre de la máquina que fingirá como servidor

NFS a su correspondiente dirección IP, puede especificar ésta en lugar del nombre.

Ejemplo:

```
mount -t nfs 192.168.1.254:/var/nfs/publico
/mnt/servidornfs
```

Podremos acceder entonces a dicho volumen remoto con solo cambiar al directorio local definido como punto de montaje, del mismo modo que se haría con un disquete o una unidad de CDROM:

```
cd /mnt/servidornfs
```

Si queremos poder montar este volumen NFS con una simple línea de comando o bien haciendo doble click en un icono sobre el escritorio, será necesario agregar la correspondiente línea en `/etc/fstab`. Ejemplo:

```
servidor.mi-red-local.org:/var/nfs/publico
/mnt/servidornfs nfs
user,exec,dev,nosuid,rw,noauto 0 0
```

La línea anterior especifica que el directorio `/var/nfs/publico/` de la máquina `servidor.mi-red-local.org` será montado en el directorio local `/mnt/servidor/nfs`, permitiéndole a los usuarios el poder montarlo, en modo de lectura y escritura y que este volumen no será montado durante el arranque del sistema. Esto último es de importancia, siendo que si el servidor no está encendido al momento de arrancar la máquina cliente, este se colgará durante algunos minutos.

6. Análisis del Intercambio de Mensajes por la Red

Cuando tenemos un ordenador local (cliente) que intenta montar un directorio de un sistema remoto (servidor) se produce el siguiente intercambio de mensajes:

	<i>Origen</i>	<i>Destino</i>	<i>protocolo</i>	<i>Información</i>
1	cliente	servidor	Tcp	
2	servidor	cliente	Tcp	
3	cliente	servidor	Tcp	
4	cliente	servidor	portmap	llamada
5	servidor	cliente	Tcp	
6	servidor	cliente	portmap	respuesta
7	cliente	servidor	Tcp	
8	servidor	cliente	Rpc	
9	cliente	servidor	Tcp	
10	cliente	servidor	Tcp	
11	servidor	cliente	Tcp	
12	cliente	servidor	Tcp	
13	cliente	servidor	mount	llamada
14	servidor	cliente	mount	respuesta
15	cliente	servidor	portmap	llamada
16	servidor	cliente	portmap	respuesta
17	cliente	servidor	Nfs	llamada
18	servidor	cliente	Nfs	respuesta
19	cliente	servidor	Nfs	llamada
20	servidor	cliente	Nfs	respuesta

Lo que está pasando en realidad es que en vez de ejecutar el procedimiento en una máquina local, RPC pasa una serie de argumentos al procedimiento en un datagrama de red. El cliente RPC crea la sesión al localizar al servidor apropiado y se envía el datagrama a un proceso en el servidor que pueda ejecutar la llamada del procedimiento remoto. En el servidor el argumento es desempaquetado, el servidor ejecuta el comando y retorna las respuestas, si hay alguna, al cliente. De vuelta en el cliente, el valor del RPC es convertido a un valor esperado por la función que lo llamo y la aplicación continua como si un procedimiento local hubiera sido llamado.

La ubicación de puertos para una sesión es manejada por el demonio *portmap*.

7. Interfaz Gráfica de Gestión

No existe ninguna interfaz gráfica, ya que no hace falta pues una vez montada las particiones remotamente podemos trabajar con los archivos montados de forma gráfica, usando el explorar de archivos que incorporan todas los entornos gráficos de Linux.

Sí es cierto que algunas versiones de Linux incorporan una mini interfaz gráfica para configurar el servidor de NFS, esto es, modificar el fichero `/etc/exports`.

Para iniciar la aplicación, seleccione **Botón de menú principal => Configuración del sistema => Configuración de servidores => Servidor NFS**, o escriba el comando `redhat-config-nfs`.

8. Ampliaciones/Mejoras del Servicio

Como ampliación podemos destacar el desarrollo de la versión 4 de NFS, que tiene un diseño muy simple, sin alejarse demasiado de las anteriores versiones cuya principal característica es que es independiente del protocolo de transporte y del sistema operativo usado. Por lo tanto es muy eficaz en red heterogéneas. Su otra gran baza es la seguridad, ya que esta versión incorpora una gran seguridad en el envío de archivos, incluso en redes de banda ancha.

Para mejorar el servicio, sobretodo a nivel de seguridad debemos de especificar algunas opciones:

La restricción de los permisos de los clientes

Cuando el administrador de una maquina monta una partición NFS, dispone de permisos de acceso de escritura, como sobre cualquier otra partición del disco local. Si la red es administrada por varios usuarios root diferentes en maquinas distintas se sugiere el uso de la opción `root_squash` en el fichero de `/etc/exports`

Esta opción elimina los privilegios de root sobre la partición montada, lo que asegura la integridad de la misma. En el marco de una exportación de `/home` impediria que el usuario root de una maquina cualquiera accediera a los directorios personales de todos los miembros de la red NFS.

Se puede usar también la opción `all_squash` que otorga a todos los usuarios los privilegios de "nobody".

El problema del UID

Utilizar la opcion `all_squash`

El problema de propiedad de los ficheros no es particularmente sensible que cuando tratamos con particiones de usuarios. Es posible esquivar el problema usando la opción `all_squash` en el montaje.

En primer lugar no montamos directamente `/home` en el conjunto de maquinas cliente, sino que el servidor debe exportarlas una a una sobre cada cliente. Hecho esto usamos la opción `all_squash` para que todas las modificaciones remotas sean consideradas como realizadas por `nobody`. Las opciones `anonuid=UID` y `anongid=GID` nos permiten reemplazar `nobody` por el UID y el GID del usuario en cuestión para que tenga acceso a su directorio personal sin problemas.

La autenticación por maquina funciona relativamente bien si los recursos personales son montados desde clientes windows donde solo hay un usuario. Es sin embargo es problemático en entornos multi-usuario.

9. Incidencias y Principales Problemas Detectados

NFS trabaja muy bien compartiendo sistemas de ficheros enteros con un gran número de máquinas conocidas de una manera muy transparente. Muchos usuarios que acceden a ficheros sobre un punto de montaje NFS pueden no estar atentos a que el sistema de ficheros que están usando no está en su sistema local. Sin embargo, esta facilidad de uso trae una variedad de potenciales problemas de seguridad.

Los siguientes puntos deben considerarse cuando exportamos un sistema de ficheros NFS en un servidor o lo montamos en un cliente. Haciendo esto minimizamos los riesgos de seguridad NFS y mejoramos la protección de sus datos y equipamiento.

Acceso al sistema

NFS controla quien puede montar y exportar sistemas de ficheros basados en la máquina que lo pide, no en el usuario que utilizará el sistema de ficheros. Las máquinas

tienen que tener los derechos para montar los sistemas de ficheros exportados explícitamente. El control de acceso no es posible para usuarios, aparte de los permisos de ficheros y directorios. En otras palabras, cuando exporta un sistema de ficheros vía NFS a una máquina remota, no sólo está confiando en la máquina a la que permite montar el sistema de ficheros, también está permitiendo a cualquier usuario que acceda a esa máquina que use su sistema de ficheros. Los riesgos de hacer esto pueden controlarse, tales como montarlo en solo lectura o cambiar a los usuarios a un ID común de usuario y grupo, pero estas soluciones impiden que el montaje sea usado de la manera originalmente prevista.

Adicionalmente, si un atacante gana el control del servidor DSN usado por el sistema que exporta el sistema de ficheros NFS, el sistema asociado con un nombre de máquina concreto o nombre de dominio totalmente cualificado, puede ser dirigido a una máquina sin autorización. En este punto, la máquina desautorizada *es* el sistema que tiene permitido montar la compartición NFS, ya que no hay intercambio de información de nombre de usuario o contraseña para proporcionar seguridad adicional al montaje NFS. Los mismos riesgos corre el servidor NIS, si los nombres de red NIS son usados para permitir a ciertas máquinas montar una compartición NFS. Usando direcciones IP en */etc/exports*, esta clase de ataques son más difíciles.

Los comodines o metacaracteres deben ser usados lo menos posible cuando garantizamos el acceso a una compartición NFS. El uso de los comodines puede permitir el acceso a sistemas que puede no saber que existen y que no deberían montar el sistema de ficheros.

Permisos de ficheros

Una vez que el sistema de ficheros es montado como lectura-escritura por una máquina remota, la protección de cada fichero compartida depende de sus permisos, y del ID de su usuario y grupo propietario. Si dos usuarios que comparten el mismo valor ID montan el mismo sistema de ficheros NFS, serán capaces de modificarse los ficheros

entre sí. Además, cualquiera que esté conectado como root en el sistema cliente, puede usar el comando `su` para convertirse en un usuario que tenga acceso a determinados archivos a través de la compartición NFS.

El procedimiento predeterminado cuando exportamos un sistema de ficheros a través de NFS es usar *root squashing* (sobreponerse a root). Esto cambia el ID de usuario de cualquiera que utilice la compartición NFS, aunque sea el root de su máquina local, al valor de la cuenta nobody del servidor. Nunca debe desactivarlo a menos que no le importe que haya múltiples usuarios con acceso de root en su servidor.

Si sólo está permitiendo a los usuarios que lean archivos de su compartición NFS, debería considerar usar la opción `all_squash`, la cual hace que todos los usuarios que accedan a su sistema de ficheros exportado tomen la ID del usuario nobody.

10. Resumen y Conclusiones

NFS permite compartir datos entre varios ordenadores de una forma sencilla y puede ser útil cuando tenemos un usuario validado en una red, éste no necesitará hacer login a un ordenador específico, ya que vía NFS, accederá a su directorio personal en la máquina en la que esté trabajando.

Pero NFS no es un protocolo demasiado eficiente y es muy lento para conexiones mediante módem. Está diseñado para redes locales, siendo muy flexible. Ofrece muchas posibilidades tanto a usuarios como a administradores.

Pero la principal desventaja de NFS es sin duda la seguridad, de hecho, ha sido apodado cariñosamente como "No File Security". Ya que NFS no utiliza un sistema de contraseñas como el que tiene SAMBA, solo una lista de control de acceso determinada por direcciones IP o nombres. Es por esto que es importante que el administrador de la red local o usuario entienda que un servidor NFS puede ser un

verdadero e inmenso agujero de seguridad si este no es configurado apropiadamente e implementado detrás de un contrafuegos o firewall.