📁 LPIC-2 / 📁 Examen 209 - Compartición de Archivos

209.3 Configurar un servidor NFS

Teoría

NFS (Network File System) es un protocolo de sistema de archivos distribuido desarrollado por Sun Microsystems que permite a los sistemas acceder a archivos a través de una red como si estuvieran almacenados localmente. Es el método estándar para compartir archivos entre sistemas Linux y Unix.

Versiones de NFS:

- **NFSv2:** La versión más antigua, menos eficiente y segura, usa principalmente UDP. Ya no es común.
- **NFSv3:** Una mejora sobre v2, más eficiente, soporta archivos más grandes, usa TCP o UDP. Aún se encuentra.
- **NFSv4:** La versión actual estándar. Es stateful (mantiene estado de la conexión), lo que mejora el manejo de firewalls. Usa por defecto y prefiere TCP en el puerto 2049. Incluye autenticación mejorada (Kerberos opcional) y otras características.

Componentes Clave del Servidor NFS (demonios del kernel):

El servidor NFS consta de varios procesos que se ejecutan en el kernel o en el espacio de usuario, gestionados por servicios systemd.

- `nfsd`: Demonio principal del servidor NFS, maneja las peticiones de los clientes.
- `rpcbind` (o `portmapper`): Convierte números de programa RPC (Remote Procedure Call) a números de puerto. Necesario para versiones antiguas de NFS (v2/v3) y servicios relacionados. Escucha en el puerto 111.
- `mountd`: Maneja las peticiones de montaje de los clientes. Necesario para NFSv3.
- `nfs-idmapd`: Mapea IDs de usuarios/grupos entre el cliente y el servidor (crucial para NFSv4 y `idmapd`).
- Otros: `lockd`, `statd` (para bloqueo de archivos y notificación de estado en NFSv3).

Implementación Básica de un Servidor NFS:

1. **Instalación del Software:**
    - **Paquete (Diferencias):**
        - **Debian/Ubuntu:** `nfs-kernel-server` (incluye los demonios del kernel y `exportfs`).
        - **Red Hat/CentOS/Fedora:** `nfs-utils` (incluye demonios, `exportfs`, y herramientas cliente como `showmount`).

- **Comando:** `sudo apt install <paquete>` o `sudo dnf install <paquete>`.

2. **Gestión de Servicios:** Asegurarse de que los servicios NFS necesarios estén corriendo.
    - **Nombre del Servicio Principal (Diferencias):**
        - **Debian/Ubuntu (moderno):** `nfs-server.service`. Controla la mayoría de los demonios kernel/user space.
        - **Red Hat/CentOS/Fedora (tradicional):** `nfs.service`. También controla los demonios.
        - **Común (Servicio RPC):** `rpcbind.service` o `portmap.service` (necesario para NFSv3 y servicios auxiliares, suele iniciarse automáticamente con el servicio nfs principal).
    - **Comandos Systemd:** `sudo systemctl enable <servicio>`, `sudo systemctl start <servicio>`, `sudo systemctl status <servicio>`.
3. **Archivo de Configuración de Comparticiones:**
    - **Ubicación:** `/etc/exports` (estándar en ambas ramas). Define qué directorios se comparten y con qué clientes y opciones.
    - **Estructura:** Cada línea define una compartición. Formato: `<directorio_a_compartir> <cliente1>(<opciones>) <cliente2>(<opciones>)...`
        - `<directorio_a_compartir>`: La ruta absoluta en el sistema de archivos local (ej: `/srv/nfs/shared`).
        - `<cliente>`: Quién tiene permiso para acceder a la compartición. Puede ser una IP única (ej: `192.168.1.10`), un rango de red (ej: `192.168.1.0/24`), un nombre de host (si es resoluble por DNS/hosts), un grupo de red (`@grupo`), o `*` (cualquiera - **PELIGROSO**).
        - `<opciones>`: Parámetros que controlan el acceso y comportamiento de la compartición.
4. **Opciones de Exportación Clave en `/etc/exports`:**
    - `ro`: Exportar el directorio en modo solo lectura.
    - `rw`: Exportar el directorio en modo lectura/escritura.
    - `sync`: Forzar la escritura de datos al disco antes de responder a las peticiones. Más seguro (los datos están en disco cuando el servidor confirma), pero más lento. Es el comportamiento por defecto en NFSv4.
    - `async`: Responder a las peticiones antes de que los datos se hayan escrito completamente al disco. Más rápido, pero menos seguro en caso de caída del servidor. Es el comportamiento por defecto en NFSv2/v3.

- `no_subtree_check`: Deshabilita la comprobación de subárbol. Puede mejorar el rendimiento, especialmente si se montan subdirectorios de la exportación. Ligeramente menos seguro si un directorio hijo se re-exporta sin el padre.
- `root_squash`: (Por defecto para root) Mapea las peticiones provenientes del usuario `root` en el cliente a un usuario anónimo en el servidor (típicamente `nobody`). Previene que el root remoto tenga privilegios de root en la compartición.
- `no_root_squash`: Deshabilita `root_squash`. Permite que el usuario `root` en el cliente acceda a los archivos como `root` en el servidor. **¡RIESGO DE SEGURIDAD!** Debe usarse con extrema precaución y solo en redes de confianza (ej: cluster privado).
- `all_squash`: Mapea *todos* los usuarios (root y no root) del cliente a un usuario anónimo en el servidor.
- `anonuid=<uid>`: Especifica el ID de usuario (UID) en el servidor al que se mapean los usuarios "squashed" (`root_squash` o `all_squash`). Por defecto, suele ser el UID de `nobody`.
- `anongid=<gid>`: Especifica el ID de grupo (GID) en el servidor al que se mapean los grupos "squashed".

5. **Aplicar Cambios en `/etc/exports`:** Después de modificar el archivo `/etc/exports`, el servidor debe volver a leer la configuración.
    - `sudo exportfs -a`: Exporta todos los directorios listados en `/etc/exports`.
    - `sudo exportfs -r`: Re-exporta todos los directorios, actualizando la lista de clientes y opciones. Es el comando más común después de editar `/etc/exports`.
    - `sudo exportfs -v`: Muestra la lista de directorios exportados y sus opciones (verbose).
    - Recargar/Reiniciar servicio: `sudo systemctl reload nfs-server` (Debian) o `sudo systemctl restart nfs` (Red Hat).
6. **Firewall:** Asegurarse de que el firewall permite el tráfico NFS necesario.
    - **NFSv4 (preferido):** Solo requiere el puerto TCP 2049.
    - **NFSv3 y servicios auxiliares:** Requiere puerto 2049 TCP/UDP, 111 TCP/UDP (rpcbind/portmapper), y puertos adicionales para mountd, statd, lockd (a menudo puertos altos y aleatorios a menos que se configuren estáticamente).
    - **Configuración de Firewall (Diferencias):**
        - **`firewalld` (Red Hat):** El servicio `nfs` permite automáticamente los puertos necesarios para NFSv3/v4. `sudo firewall-cmd --add-service=nfs --permanent` y `sudo firewall-cmd --reload`.
        - **`ufw` (Debian/Ubuntu):** Puede añadir una regla predefinida `NFS`. `sudo ufw allow NFS`.
        - **`iptables`:** Añadir reglas explícitas para los puertos relevantes.

7. **Versión de NFS:** El servidor y el cliente negocian la mejor versión de NFS que ambos soportan. Puedes forzar una versión específica en el cliente (Ej. 209.4). NFSv4 es más fácil con firewalls.
8. **Mapeo de Usuarios y Grupos:** Para que los permisos de archivo funcionen correctamente en NFS (sin `all_squash`), los UID y GID deben coincidir entre el servidor y el cliente. Esto se puede lograr usando un servicio de directorio centralizado como LDAP o asegurándose de que los usuarios/grupos relevantes tienen los mismos IDs en ambos sistemas manualmente. NFSv4 usa `nfs-idmapd` para ayudar con este mapeo, a menudo basado en nombres de usuario/grupo en formato `user@domain`.