Le protocole NFS

I. Présentation

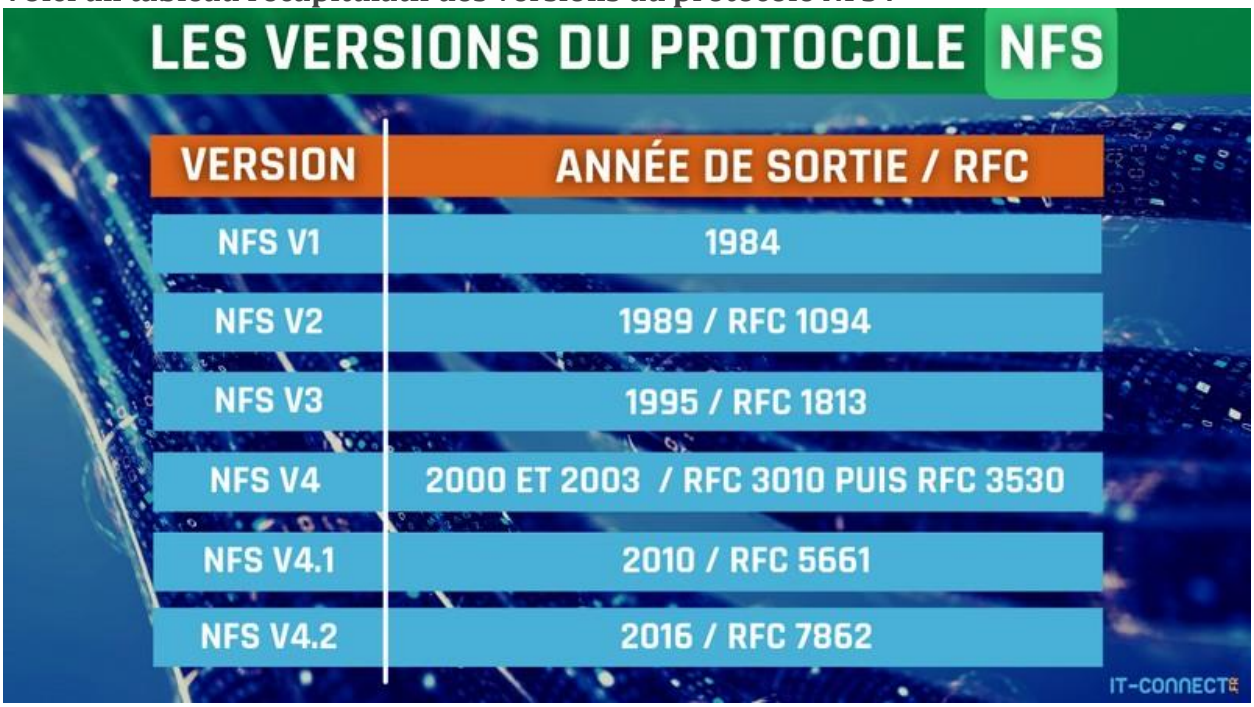
le protocole NFS, un protocole de communication qui sert à effectuer des transferts de fichiers, dans le même esprit que le protocole SMB.

Le NFS, ***N**etwork **F**ile **S**ystem*, que l'on pourrait traduire par "*Système de fichiers en réseau*", est **un protocole de transferts de fichiers par le réseau principalement utilisé sur les systèmes Linux/Unix**, même s'il est compatible avec Windows et MacOS. Dans la pratique, l'utilisateur, à partir de son ordinateur, va pouvoir accéder à des fichiers stockés sur un serveur distant, à l'aide du protocole NFS qui fonctionne selon le mode client/serveur.
Contrairement au SMB qui est un protocole propriétaire Microsoft, le protocole NFS est libre. Il s'installe sur Linux par l'intermédiaire d'un paquet dédié, et il est pris en charge par des solutions comme VMware ESXi pour connecter une banque de données, mais aussi par les NAS présent sur le marché (Synology, Asustor, Qnap, etc.).
Le protocole NFS est majoritairement utilisé pour connecter un espace de stockage entre un serveur NFS et un ou plusieurs serveurs, par exemple pour stocker des sauvegardes ou héberger des machines virtuelles. Il est trop complexe à mettre en oeuvre pour la mise en place d'un véritable serveur de fichiers, où là on va préférer le protocole SMB.

II. Les différentes versions de NFS

Développé par Sun Microsystems en 1984, le protocole NFS existe depuis plusieurs dizaines d'années comme de nombreux autres protocoles. Les versions 1 et 2 du protocole NFS fonctionnaient à l'aide de connexion UDP, tandis que le TCP (en mode *stateless*, c'est-à-dire sans état) a fait son apparition avec le NFS v3.
Les premières versions, que ce soit NFS v1, NFS v2 ou NFS v3 n'étaient pas sécurisées : à l'époque, la sécurité n'était pas une priorité. Il y avait également des limitations sur la taille des paquets (8 Ko) et la taille maximale d'un fichier transférable (2 Go) avant l'arrivée de NFS v3.
C'est en **2000**, avec la sortie de **NFS v4 que le protocole a fortement évolué pour intégrer des fonctionnalités liées à la sécurité.** Il a tellement évolué, que le NFS v4 marque une véritable rupture vis-à-vis des versions précédentes (et cela n'est pas sans conséquence, nous en reparlerons). Depuis, le protocole NFS a eu le droit à plusieurs mises à jour, en version 4.1 en 2010 et un peu plus récemment, en 2016, en version 4.2.
Le protocole NFS v4 prend en charge complètement la sécurité avec l'authentification via Kerberos et il s'appuie sur des connexions TCP avec suivi de l'état (*stateful*).

Voici un tableau récapitulatif des versions du protocole NFS :



Les différentes versions du protocole NFS

III. Compatibilité entre les versions de NFS

Il n'y a pas de rétrocompatibilité entre le NFS v4 (et supérieur) et les versions précédentes, y compris avec le NFS v3. Les différences sont trop nombreuses, tant sur l'aspect de la sécurité, mais aussi le fonctionnement des connexions : *TCP stateful* avec NFS v4.

Autrement dit, si le serveur fonctionne uniquement avec NFS v4, le client ne pourra pas utiliser NFS v3. Si les deux effectuent la connexion en NFS v3, ce sera bon. Généralement, les serveurs NFS sont capables de gérer plusieurs versions. Par contre, si vous souhaitez utiliser l'authentification Kerberos, ce sera obligatoirement en NFS v4, seule version à prendre en charge cette sécurité.

IV. Quel est le port utilisé par le protocole NFS ?

Le protocole NFS s'appuie sur un seul port pour fonctionner : le port 2049. Il s'agit du **port d'écoute d'un serveur NFS**.

Néanmoins, je tiens à préciser que c'est **à partir de NFS v4 que le protocole utilise un seul port.** Les anciennes versions utilisaient le **port 2049**, mais aussi le **port 111** correspondant au service "*portmap*" (utile pour orienter les requêtes RPC vers le bon service).

Une bonne nouvelle pour la configuration de votre pare-feu si vous utilisez les dernières versions de NFS, car il n'y aura qu'un seul port à autoriser, comme pour d'autres protocoles.

V. Créer un partage NFS sous Debian (Linux)

Après cette introduction théorique, nous allons passer à la pratique. Pour cela, nous allons utiliser deux machines sous Linux (Debian 11), mais vous pouvez utiliser d'autres distributions.

- **Un serveur Debian** qui jouera le rôle de **serveur NFS**, avec un partage : */srv/partagenfs*
  - Adresse IP : 192.168.100.121/24
- **Un poste client Debian** qui jouera le rôle de **client NFS**
  - Adresse IP : 192.168.100.120/24

Commençons par préparer notre serveur. Ensuite, nous allons monter ce partage NFS sur le poste client et nous finirons par une analyse de trames avec *Tcpdump*.
Note : pour cette introduction au NFS, nous n'allons pas mettre en place l'authentification Kerberos car cela complexifie la procédure.

A. Installation du paquet

Sur le serveur Debian, mettez à jour le cache des paquets :

```
sudo apt-get update
```

Ensuite, installez le paquet "*nfs-kernel-server*" :

```
sudo apt-get install nfs-kernel-server
```

```
root@debian-11:~# apt-get install nfs-kernel-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  keyutils libevent-2.1-7 libnfsidmap2 nfs-common rpcbind
Paquets suggérés :
  open-iscsi watchdog
Les NOUVEAUX paquets suivants seront installés :
  keyutils libevent-2.1-7 libnfsidmap2 nfs-common nfs-kernel-server rpcbind
0 mis à jour, 6 nouvellement installés, 0 à enlever et 23 non mis à jour.
Il est nécessaire de prendre 682 ko dans les archives.
Après cette opération, 2□028 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] O
Réception de :1 http://ftp.fr.debian.org/debian bullseye/main amd64 rpcbind amd64 1.2.5-9 [51,4 kB]
Réception de :2 http://ftp.fr.debian.org/debian bullseye/main amd64 keyutils amd64 1.6.1-2 [52,8 kB]
Réception de :3 http://ftp.fr.debian.org/debian bullseye/main amd64 libevent-2.1-7 amd64 2.1.12-stab
Réception de :4 http://ftp.fr.debian.org/debian bullseye/main amd64 libnfsidmap2 amd64 0.25-6 [32,6
Réception de :5 http://ftp.fr.debian.org/debian bullseye/main amd64 nfs-common amd64 1:1.3.4-6 [232
Réception de :6 http://ftp.fr.debian.org/debian bullseye/main amd64 nfs-kernel-server amd64 1:1.3.4-
```

Installation serveur NFS sous Debian

Nous allons configurer le serveur NFS pour qu'il démarre automatiquement avec le système :

```
sudo systemctl enable nfs-server.service
```

Le paquet est installé, passons à l'étape suivante.

B. Déclarer un partage NFS /etc/exports

Commençons par créer le partage :

mkdir /srv/partagenfs

Puis, on applique les droits sur le partage (à adapter selon vos besoins) :

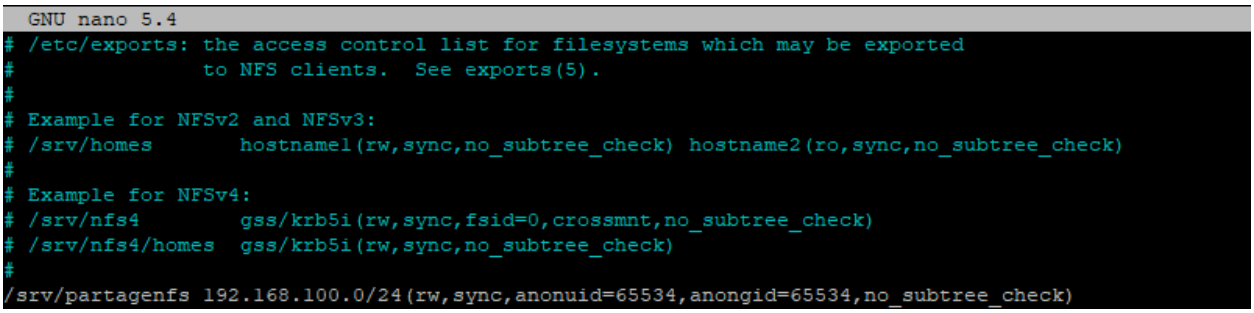
```
chown nobody:nogroup /srv/partagenfs/

chmod 755 /srv/partagenfs/
```

Pour déclarer les partages NFS, il faut s'appuyer sur le fichier "*/etc/exports*". C'est dans ce fichier que nous allons déclarer notre fichier "*/srv/partagenfs*". Editez le fichier :

```
sudo nano /etc/exports
```

Dans ce fichier, voici la ligne à ajouter pour déclarer notre partage :

```
/srv/partagenfs
192.168.100.0/24(rw,sync,anonuid=65534,anongid=65534,no_subtree_check)
```



Quelques explications s'imposent pour bien comprendre

- **/srv/partagenfs** : le chemin local du dossier à partager
- **192.168.100.0/24** : l'adresse IP ou le réseau à autoriser, si vous souhaitez autoriser seulement une adresse IP spécifique, précisez cette adresse IP
- **()** : les options pour le partage
- **rw** : partage accessible en lecture et écriture, à remplacer par "**ro**" pour la lecture seule
- **sync** : écrire les données et les vérifier avant de répondre à la requête suivante : plus lent, mais plus fiable vis-à-vis des corruptions de données. L'autre mode est "*async*".
- **anonuid** : ID de l'utilisateur à utiliser pour les connexions anonymes (*65534 = nobody*)
- **anongid** : ID du groupe à utiliser pour les connexions anonymes (*65534 = nogroup*)
- **no_subtree_check** : désactiver la vérification des sous-dossiers, recommandé pour des raisons de fiabilité

Il est à noter que l'on peut déclarer plusieurs adresses IP ou réseaux avec des autorisations différentes (ou identiques). Par exemple :

```
/srv/partagenfs
192.168.100.0/24(rw,sync,anonuid=65534,anongid=65534,no_subtree_check)
10.10.10.0/24(rw,sync,anonuid=65534,anongid=65534,no_subtree_check)
```

Enregistrez le fichier et fermez-le. Pour que la configuration soit prise en compte, il faut utiliser *exportfs*, comme ceci :

```
exportfs -a
```

Il est à noter que pour **stopper et purger les partages NFS**, il faut exécuter la commande suivante (à faire avant la commande précédente pour actualiser) :

```
exportfs -ua
```

La commande ci-dessous permet d'afficher la liste des partages NFS sur l'hôte précisé, en l'occurrence notre machine Debian elle-même.

```
showmount -e 192.168.100.121
```

```
root@debian-11:~# showmount -e 192.168.100.121
Export list for 192.168.100.121:
/srv/partagenfs 10.10.10.0/24,192.168.100.0/24
```

Exemple de la commande showmount nfs

Sur la copie d'écran ci-dessus, on peut voir le nom du partage "*/srv/partagenfs*" ainsi que les deux sous-réseaux autorisés comme dans l'exemple avec deux adresses IP mentionnées précédemment.
La commande ci-dessous est à connaître également pour lister les services en écoute via RPC. **On peut voir le service "nfs" avec différentes versions, notamment NFS v3 et NFS v4.** On voit également la présence de *portmapper* sur le port 111 (service non spécifique à NFS, mais utilisé par certaines versions).

```
rpcinfo -p
```

```
root@debian-11:~# rpcinfo -p
   program vers proto   port  service
    100000    4   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp  41278  mountd
    100005    1   tcp  34225  mountd
    100005    2   udp  36289  mountd
    100005    2   tcp  32791  mountd
    100005    3   udp  39225  mountd
    100005    3   tcp  44325  mountd
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049
    100003    3   udp   2049  nfs
    100227    3   udp   2049
    100021    1   udp  37581  nlockmgr
    100021    3   udp  37581  nlockmgr
    100021    4   udp  37581  nlockmgr
    100021    1   tcp  38809  nlockmgr
    100021    3   tcp  38809  nlockmgr
    100021    4   tcp  38809  nlockmgr
```

Aperçu de NFS avec rpcinfo

Le partage NFS est prêt, passons sur le poste client.

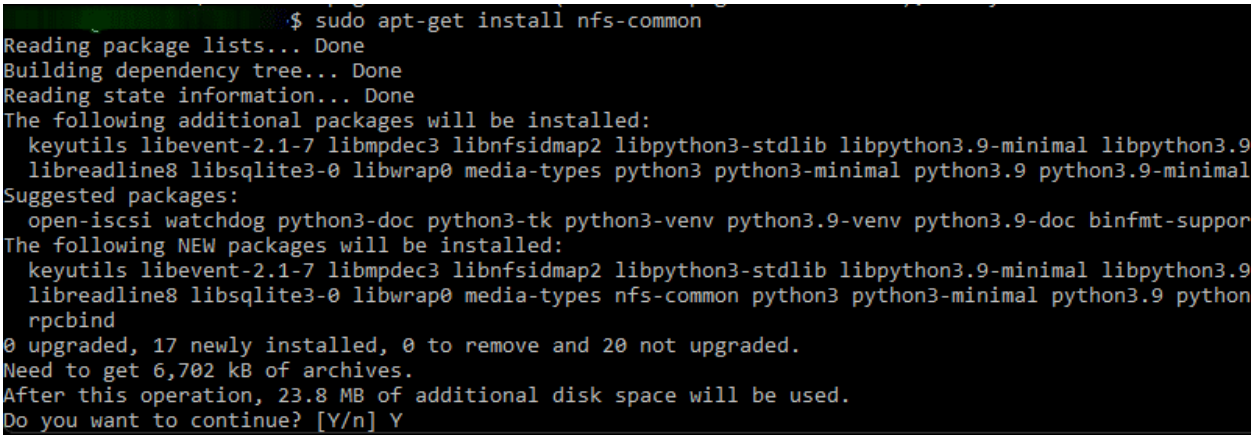
C. Connexion au partage NFS depuis Linux

Sur le poste Debian, mettez à jour le cache des paquets :

```
sudo apt-get update
```

Puis, installez le paquet "*nfs-common*" afin de pouvoir monter le partage NFS sur l'hôte local.

```
sudo apt-get install nfs-common
```



Installation d'un client NFS sous Debian

Maintenant, nous allons pouvoir monter manuellement le partage NFS sur notre machine.

Pour monter le partage, nous devons créer un dossier local qui servira de point de montage. Créez ce dossier :

```
mkdir /media/partagenfs
```

Ensuite, la commande "mount" va nous permettre de monter notre partage NFS. Ce qui donne :

```
mount -t nfs4 192.168.100.121:/srv/partagenfs /media/partagenfs/
```

Quelques explications :

- **-t nfs4** : utiliser NFS v4 pour monter ce partage. Pour utiliser une version antérieure, il faut spécifier "*-t nfs*".
- **192.168.100.121:/srv/partagenfs** : partage NFS à monter, sous la forme *<hote>:<chemin-partage>*
- **/media/partagenfs/** : point de montage local

À la suite de cette commande, vous devriez pouvoir créer un fichier sur le partage :

```
touch /media/partagenfs/monfichier.txt
```

Sur le serveur NFS, on peut voir les différents fichiers créés :

```
root@debian-11:~# ls -l /srv/partagenfs/
total 0
-rw-r--r-- 1 nobody nogroup 0 11 oct.  13:59 monfichier2.txt
-rw-r--r-- 1 nobody nogroup 0 11 oct.  14:18 monfichier3.txt
-rw-r--r-- 1 nobody nogroup 0 11 oct.  13:59 monfichier.txt
```

Il faut savoir qu'avec la commande "mount", le montage sera temporaire, c'est-à-dire que si l'on redémarrer le poste client, le partage ne sera pas monté automatiquement.

Démontez le partage et nous allons le monter différemment :

```
umount /media/partagenfs
```

Ensuite, modifiez la table de montage "*/etc/fstab*" :

```
sudo nano /etc/fstab
```

Spécifiez la ligne suivante pour le monter automatiquement, toujours via NFS v4 :

```
192.168.100.121:/srv/partagenfs /media/partagenfs nfs4 defaults,user,exec 0
0
```

Enregistrez puis exécutez la commande ci-dessous pour charger le contenu du fichier "*/etc/fstab*" et monter notre partage.

```
sudo mount -a
```

Voilà, vous devriez pouvoir accéder au partage NFS de la même manière qu'en le montant avec *mount*.
Toujours sur le poste client, exécutez la commande ci-dessous pour obtenir des informations sur l'activité NFS de la machine.

```
nfsstat -m
```

Cette commande me retourne :

```
/media/partagenfs from 192.168.100.121:/srv/partagenfs

Flags:
rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,hard,proto=tcp,t
imeo=600,retrans=2,sec=sys,clientaddr=192.168.100.120,local_lock=none,addr
=192.168.100.121
```

On peut voir que l'accès est bien en lecture/écriture (*rw*), mais aussi que l'on utilise la version NFS 4.2 (*vers=4.2*) et que l'on s'appuie sur la sécurité système (*sec=sys*) tandis que l'on aurait "*sec=krb5*" si l'on utilisait Kerberos.
Enfin, si l'on regarde les connexions actives sur le poste client Debian ou sur le serveur Debian, on peut voir qu'il y a une connexion établie TCP (ESTABLISHED) sur le port NFS, c'est-à-dire 2049.

```
netstat -petula | grep "nfs"
```

```
root@debian-11:~# netstat -petula | grep "nfs"
tcp        0      0 0.0.0.0:nfs             0.0.0.0:*               LISTEN      root
tcp        0      0 192.168.100.121:nfs     192.168.100.120:958     ESTABLISHED nobody
tcp6       0      0 [::]:nfs                [::]:*                  LISTEN      root
udp        0      0 0.0.0.0:nfs             0.0.0.0:*                           root
udp6       0      0 [::]:nfs                [::]:*                              root
```

D. Capture des paquets NFS avec TCPDUMP

Terminons ce tutoriel par **une capture de paquets réseau avec l'utilitaire TCPDUMP**. Cela va vous faire manipuler cet outil fort pratique afin de vérifier que les connexions NFS v4 fonctionnent bien sur le port 2049.

Sur le serveur Debian, installez l'outil "*tcpdump*" :

```
sudo apt-get install tcpdump
```

Ensuite, nous allons lancer une capture TCPDUMP en filtrant sur deux numéros de ports : 2049 et 111 :

```
tcpdump port 2049 or port 111
```

Pendant ce temps, depuis le poste client, créez de l'interaction avec le partage NFS. Par exemple, en lisant le contenu du répertoire :

```
ls /media/partagenfs
```

Côté serveur

```
root@debian-11:~# tcpdump port 2049,111
tcpdump: can't parse filter expression: syntax error
root@debian-11:~# tcpdump port 2049 or port 111
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens192, link-type EN10MB (Ethernet), snapshot length 262144 bytes

14:18:26.240775 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [P.], seq 2101767541:2101767717,
36 172 getattr fh 0,2/53
14:18:26.240937 IP 192.168.100.121.nfs > 192.168.100.120.738: Flags [P.], seq 1:253, ack 176, win 50
NON 3 ids 0/1292461153 sz 3534802317
14:18:26.241008 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [.], ack 253, win 501, options [
14:18:26.247614 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [P.], seq 176:280, ack 253, win
2/44
14:18:26.247781 IP 192.168.100.121.nfs > 192.168.100.120.738: Flags [P.], seq 253:301, ack 280, win
[|nfs]
14:18:26.247847 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [.], ack 301, win 501, options [
14:18:26.247896 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [P.], seq 280:376, ack 301, win
14:18:26.251067 IP 192.168.100.121.nfs > 192.168.100.120.738: Flags [P.], seq 301:349, ack 376, win
[|nfs]
14:18:26.251129 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [.], ack 349, win 501, options [
14:18:26.251764 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [F.], seq 376, ack 349, win 501,
14:18:26.251775 IP 192.168.100.121.nfs > 192.168.100.120.738: Flags [F.], seq 349, ack 377, win 501,
14:18:26.251821 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [.], ack 350, win 501, options [
14:18:27.607716 IP 192.168.100.120.733 > 192.168.100.121.nfs: Flags [S], seq 4252914668, win 64240,
14:18:27.607750 IP 192.168.100.121.nfs > 192.168.100.120.733: Flags [S.], seq 1679536233, ack 425291
14:18:27.607821 IP 192.168.100.120.733 > 192.168.100.121.nfs: Flags [.], ack 1, win 502, options [no
14:18:27.607857 IP 192.168.100.120.733 > 192.168.100.121.nfs: Flags [P.], seq 1:45, ack 1, win 502,
14:18:27.607865 IP 192.168.100.121.nfs > 192.168.100.120.733: Flags [.], ack 45, win 509, options [n
14:18:27.607952 IP 192.168.100.121.nfs > 192.168.100.120.733: Flags [P.], seq 1:29, ack 45, win 509,
14:18:27.607991 IP 192.168.100.120.733 > 192.168.100.121.nfs: Flags [.], ack 29, win 502, options [n
14:18:27.608099 IP 192.168.100.120.733 > 192.168.100.121.nfs: Flags [P.], seq 45:277, ack 29, win 50
42
```

Capture de paquets NFS avec tcpdump

On peut voir de nombreuses lignes avec la mention "nfs" pour indiquer que le serveur utilise le port NFS (2049) pour communiquer avec le client. **Quant au port 111, il n'apparaît à aucun moment.**

```
14:18:26.240775 IP 192.168.100.120.738 > 192.168.100.121.nfs: Flags [P.], seq 2101767541:2101767717, ack 108469392, win 501, options [nop,nop,TS val 759855997 ecr 3348659237], length 176: NFS request xid 2540827736 172 getattr fh 0,2/53
```

Si vous avez le doute, il suffit d'arrêter cette capture et d'en lancer une nouvelle en spécifiant seulement le port "111" et vous verrez que la capture restera vide.