

SERVIDORES WEB DE ALTA DISPONIBILIDAD (2021-2022)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 6

Pedro Antonio Mayorgas Parejo

9 de junio de 2022

Índice

1	Servidor NFS	3
1.1	Fichero exports	5
1.2	Configuración del cliente	8
1.3	fstab filesystem table	9
1.4	Iptables	9
1.4.1	Reglas de iptables	12
2	Bibliografía	21

1. Servidor NFS

NFS Network File System: Protocolo que permite el acceso remoto a un sistema de archivos a través de la red. Es un módulo del kernel de Linux.

Para poder trabajar con NFS necesitamos instalar los siguientes paquetes:

```
1 # Instalacion del servidor
2 apt install nfs-kernel-server portmap
3 # Instalacion del cliente
4 sudo apt-get install nfs-common
```

Portmap: Demonio de asignación de puertos dinámico para servicios RPC, como NFS. Es complicado de asegurar y se usa en las implementaciones de NFSv3 para abajo. Posee poca seguridad y está siendo Deprecated (obsoleto), por la versión de NFSv4.

RPC Remote Procedure Call - paquete RPCBIND: Estándar de UNIX para servicios remotos. Los servicios que lo utilicen, se registran en un directorio conocido como portmapper (Asociador de Puertos - Mapeo de Puertos). Un cliente que desee realizar una consulta NFS primero debe dirigirse al portmapper (en el puerto 111 TCP/UDP). Cuando le devuelva la respuesta el portmapper, le indica el puerto del servicio NFS.

```
debian@m1-pedroamp:~$ sudo apt-get install nfs-kernel-server portmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'rpcbind' instead of 'portmap'
The following additional packages will be installed:
  keyutils libevent-2.1-7 libnfsidmap2 nfs-common
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-2.1-7 libnfsidmap2 nfs-common nfs-kernel-server rpcbind
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 682 kB of archives.
After this operation, 2028 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Figura 1: Instalación del servicio NFS en m1-pedroamp.

```
debian@m2-pedroampcopy:~$ sudo apt-get install nfs-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  keyutils libevent-2.1-7 libnfsidmap2 rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-2.1-7 libnfsidmap2 nfs-common rpcbind
0 upgraded, 5 newly installed, 0 to remove and 1 not upgraded.
Need to get 557 kB of archives.
After this operation, 1677 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Figura 2: Instalación del cliente NFS en m2-pedroamp.

Mi servicio de NFS en Debian GNU/Linux. Está activado por defecto. No tengo que realizar ninguna configuración adicional, pero en el caso de que esté desactivado debemos poner:

```
1 sudo systemctl enable nfs-kernel-server.service
2 sudo systemctl start nfs-kernel-server.service
```

```
3 sudo systemctl status nfs-kernel-server.service
```

1.1. Fichero exports

Es el fichero indicado para la configuración del servidor NFS. Lista los directorios que sí están disponibles a lo largo de la red (Exportables). Por cada compartición por NFS, se puede indicar por granularidad a quién se le permite (host) hacer el qué.

Una configuración avanzada sería:

```
1 # Configuración para permitir a un host en concreto escribir en el
   directorio exportado por NFS.
2 /srv/copy 192.168.122.119(rw, sync, no_subtree_check, root_squash)
3
4 # Configuración de solo lectura en todos los demás hosts de la red.
5 /srv/copy 192.168.122.0/24(ro, sync, no_subtree_check, root_squash)
6
7 # En la configuración real para que se exporte el mismo directorio a
   determinados hosts o subredes concretas. Se indica como sigue
8 /srv/copy 192.168.122.119(rw, sync, no_subtree_check, root_squash)
   192.168.122.0/24(ro, sync, no_subtree_check, root_squash)
```

1. **sync:** Es una opción que sigue el protocolo de NFS que indica que responde a las peticiones solamente si los cambios se han aplicado en el almacenamiento de disco. Es una manera segura, pero un poco más lenta.
2. **async:** Es una opción que se salta el protocolo de NFS. Que permite al servidor, que pueda responder peticiones antes de que cualquier cambio de una petición anterior haya podido ser registrada (commit), en el almacenamiento. Esta opción mejora el rendimiento, pero si el servidor es propenso a tener una caída registrará datos corruptos.
3. **no_subtree_check:** Desactiva la comprobación del subarbol. Tiene una implicación de seguridad leve, pero mejora la fiabilidad. Lo que hace es desactivar una comprobación en la que los subdirectorios del sistema y sus archivos son tratados como valores binarios opacos. No como por ruta o nombre como suele ser habitual en un sistema de ficheros estándar. Entonces el servidor con esta opción puesta, no va a comprobar que cada petición de NFS es un fichero localizado en el área que está exportando (Subtree_check). Entonces un atacante malicioso puede manipular al sistema NFS para obtener cierta información que no está en el fichero de NFS.
¿Entonces por qué no ponerlo por defecto?

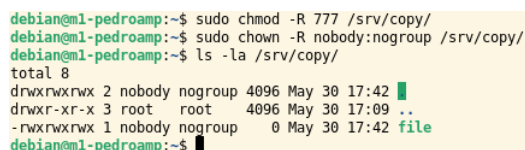
- a) Los efectos causados por el subtree_check es que se rompen el cumplimiento del protocolo NFS, es decir que cuando cualquier cliente solicite un fichero en concreto o un directorio, se le entrega la información del directorio padre al que pertenece. Si cambia de nombre, o es movido ese fichero a otro directorio. Esa información que en concurrencia haya sido solicitada por otro cliente, está obsoleta causando que cuando quiera escribir o leer en el, aparezca como errores ESTALE (Estallido), es decir que el cliente sigue creyendo que el fichero que ha sido movido sigue en el mismo lugar produciéndose una pérdida de información crítica. Debido a que indica que está escribiendo en una ruta

determinada y se corrompe la información. Entonces por eso se usa el modo `no_subtree_check`, en el cual se trata como binario uniforme.

- b) Para solucionar los problemas de seguridad del `no_subtree_check`, se recomienda en la documentación particionar o tener un disco completo con un sistema de ficheros entero aislado del sistema principal.
4. **squashing Aplastamiento de permisos.** Es necesario para evitar que por ejemplo el usuario root de un sistema extraño al de NFS pueda leer, escribir ficheros con privilegios elevados.
- a) **all_squash:** Todos los usuarios root y no-root. Se les quita el UID y GID, asignándoles uno por defecto con **anonuid y anongid**. Esto quiere decir que su grupo de permisos son `nobody:nogroup` en los cuales se pueden leer o escribir si NFS lo permite.
 - b) **root_squash:** Los usuarios root solamente, se les quita su peligroso UID y GID 0. Junto con las opciones **anonuid y anongid**. Podemos hacer que dichos usuarios con privilegios elevados en el sistema ajeno a NFS, pueda acceder y escribir bajo los mismos permisos que un usuario estándar.
 - c) **no_root_squash:** Los usuarios root acceden como tales. Pero esto es peligroso.

La carpeta de /srv Es una carpeta indicada por el estándar general de POSIX de UNIX para indicar las rutas por defecto por donde se comparte información. Para crear la carpeta exportable hemos indicado los siguiente:

```
1 # Creacion de la carpeta exportada
2 sudo mkdir /srv/copy
3 # Asignando propietarios de la carpeta a nobody:nogroup
4 sudo chown -R nobody:nogroup
5 # Asignando permisos de lectura y escritura a otros para que puedan
  trabajar en el directorio
6 sudo chmod -R 777 /srv/copy/
```



```
debian@ml-pedroamp:~$ sudo chmod -R 777 /srv/copy/
debian@ml-pedroamp:~$ sudo chown -R nobody:nogroup /srv/copy/
debian@ml-pedroamp:~$ ls -la /srv/copy/
total 8
drwxrwxrwx 2 nobody nogroup 4096 May 30 17:42 .
drwxr-xr-x 3 root root 4096 May 30 17:09 ..
-rwxrwxrwx 1 nobody nogroup 0 May 30 17:42 file
debian@ml-pedroamp:~$
```

Figura 3: Creación del directorio de exportación. Falta el `chmod`

La configuración en el lado del servidor en `/etc/exports`.

```

debian@ml-pedroamp:~$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/srv/copy 192.168.122.119(rw,sync,no_subtree_check,all_squash) 192.168.122.0/24(ro,sync,no_subtree_check,all_squash)

```

Figura 4: Fichero de configuración /etc/exports.

A continuación reiniciamos el servicio de NFS para poder aplicar los cambios.

- 1 sudo systemctl restart nfs-server.service
- 2 sudo systemctl status nfs-server.service
- 3 sudo showmount -e localhost

showmount Es una utilidad que permite mostrar todos los directorios que exporta el servicio de NFS. En el caso en concreto del propio servidor se lee en localhost.

```

debian@ml-pedroamp:~$ sudo systemctl status nfs-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
   Active: active (exited) since Mon 2022-05-30 17:27:44 UTC; 7s ago
     Process: 2328 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
     Process: 2329 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
    Main PID: 2329 (code=exited, status=0/SUCCESS)
       CPU: 9ms

May 30 17:27:43 ml-pedroamp systemd[1]: Starting NFS server and services...
May 30 17:27:44 ml-pedroamp systemd[1]: Finished NFS server and services.
debian@ml-pedroamp:~$ sudo showmount -e localhost
Export list for localhost:
/srv/copy 192.168.122.0/24
debian@ml-pedroamp:~$ █

```

Figura 5: Cambios aplicados y comprobando que se exporte el directorio.

1.2. Configuración del cliente

La instalación del cliente se ha seguido al principio de la sección. Aquí vamos a ver los parámetros manuales y los automáticos de montaje del directorio exportado de NFS.

El cliente que se va a mostrar es m2-pedroamp, que tiene permisos de escritura. Recordemos que la subred entera es la que tiene de solo lectura y se tratará de probar después de m2 con m3 si puede escribir.

```
1 # Comprobamos que el directorio a importar este disponible
2 sudo showmount -e 192.168.122.70
3 # Creamos la carpeta
4 mkdir $HOME/importcopy
5 # Comando de mount para montar el directorio
6 sudo mount -t nfs -o auto,noexec,user,rw,sync,nosuid 192.168.122.70:
   /srv/copy $HOME/importcopy/
7 # Ver si se ha montado correctamente la unidad.
8 sudo mount | tail -n 1
```

Las opciones de mount son:

- **auto:** Especifica que la partición o sistema de archivos debe ser automontada si ejecutamos mount -a. Tiene más efecto en el fichero /etc/fstab.
- **noexec:** Su opción contraria **exec** indica que los ficheros del sistema de ficheros, pueden ser ejecutables. La opción fijada es que **NO** se permite ejecutar los ficheros ejecutables en el NFS.
- **user:** Esta opción permite al usuario montar/desmontar las particiones. Su opción contraria **nouser** implica que solo el usuario root es el que puede montar/desmontar las particiones del sistema de ficheros indicado.
- **sync:** Especifica que la entrada y salida desde el sistema de ficheros. Debe ser tratada de manera síncrona. La opción **async**, es asíncrona, es decir no fuerza a escribir desde el mismo momento en el que escribes algo en el sistema de ficheros, si no que pasa a ser encolado o está escribiéndose más lentamente en background.
- **rw (o ro):** El montaje del sistema de ficheros especificado, se realiza en modo de escritura o solo lectura (read only).
- **nosuid:** Evita la ejecución privilegiada a través del bit SUID de los ficheros.


```
debian@m1-pedroamp:~$ ls -la /srv/copy/
total 12
drwxr-xr-x 2 nobody nogroup 4096 May 31 11:00
drwxr-xr-x 3 root root 4096 May 30 17:09
-rw-r--r-- 1 nobody nogroup 17 May 31 11:00 m2-pedroamp
debian@m1-pedroamp:~$

debian@m2-pedroampcopy:~$ sudo showmount -e 192.168.122.70
Export list for 192.168.122.70:
/srv/copy 192.168.122.0/24
debian@m2-pedroampcopy:~$ sudo mount -t nfs -o auto,noexec,user,rw,sync,nosuid 192.168.122.70:/
/srv/copy $HOME/importcopy
debian@m2-pedroampcopy:~$ sudo mount | tail -n 1
192.168.122.70:/srv/copy on /home/debian/importcopy type nfs4 (rw,nosuid,nodev,noexec,relatime,
sync,vers=4.2,rsz=524288,wsz=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,c
lientaddr=192.168.122.119,local_lock=none,addr=192.168.122.70,user)
debian@m2-pedroampcopy:~$ sudo touch importcopy/m2-pedroamp
debian@m2-pedroampcopy:~$ echo "Hola m3-pedroamp" > importcopy/m2-pedroamp
debian@m2-pedroampcopy:~$

debian@m3-pedroamp:~$ sudo showmount -e 192.168.122.70
Export list for 192.168.122.70:
/srv/copy 192.168.122.0/24
debian@m3-pedroamp:~$ sudo mount -t nfs -o auto,noexec,user,rw,sync,nosuid 192.168.122.70:/srv/copy $HOME/importcopy
debian@m3-pedroamp:~$ sudo mount | tail -n 1
192.168.122.70:/srv/copy on /home/debian/importcopy type nfs4 (rw,nosuid,nodev,noexec,relatime,
sync,vers=4.2,rsz=524288,wsz=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,
clientaddr=192.168.122.57,local_lock=none,addr=192.168.122.70,user)
debian@m3-pedroamp:~$ touch importcopy/m3-pedroamp
touch: cannot touch 'importcopy/m3-pedroamp': Read-only file system
debian@m3-pedroamp:~$ cat importcopy/m2-pedroamp
Hola m3-pedroamp
debian@m3-pedroamp:~$
```

Figura 6: Montaje de la unidad de NFS y creando ficheros en ella.

Podemos ver que en m3-pedroamp no tiene permisos de escritura indicados en el fichero de configuración `/etc/exports` a pesar de indicarle con `mount` que sí lo tiene. Podemos ver que el UID del root ha sido aplastado (squash). Porque podemos ver que el usuario puede escribir en el fichero previamente creado por el usuario root de su sistema. Así como en m3 con otro usuario tampoco tiene problemas para leerlo.

1.3. fstab filesystem table

Ahora vamos a hacerlo permanente con `fstab` el montaje en m2-pedroamp. No hay mucha variación con el comando de `mount`, las opciones son las mismas. Quedaría como sigue.

```
GNU nano 5.4 /etc/fstab
# /etc/fstab: static file system information
#
# UUID=97bb8630-cb1a-4917-9d75-222c701a2cc7 / ext4 rw,discard,errors=remount-ro,x-systemd.growfs 0 1
UUID=7FFD-88D9 /boot/efi vfat defaults 0 0
192.168.122.70:/srv/copy /home/debian/copy nfs auto,noexec,user,rw,sync,nosuid 0 0
```

Figura 7: Fichero de configuración de sistemas de ficheros FSTAB.

```
peter@mark-6:~$ ssh debian@192.168.122.119
Linux m2-pedroampcopy 5.10.0-14-amd64 #1 SMP Debian 5.10.113-1 (2022-04-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 31 11:19:52 2022 from 192.168.122.1
debian@m2-pedroampcopy:~$ sudo mount | grep nfs
192.168.122.70:/srv/copy on /home/debian/copy type nfs4 (rw,nosuid,nodev,noexec,relatime,
sync,vers=4.2,rsz=524288,wsz=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,
clientaddr=192.168.122.119,local_lock=none,addr=192.168.122.70,user)
debian@m2-pedroampcopy:~$ sudo tail -n 1 /etc/fstab
192.168.122.70:/srv/copy /home/debian/copy nfs auto,noexec,user,rw,sync,nosuid 0 0
debian@m2-pedroampcopy:~$
```

Figura 8: Fstab automontando el disco NFS.

1.4. Iptables

Nuestra configuración de `iptables`, no va a funcionar con los puertos aleatorios que crea `RPC`. Debido a eso tenemos que configurar en el servidor un modo en el cual los puertos de `RPC` sean conocidos, para poder permitir su salida. Para ello nos vamos al fichero `/etc/default/nfs-kernel-server`. Donde fijamos el puerto de `RPC` como sigue:

```

GNU nano 5.4 /etc/default/nfs-kernel-server
# Number of servers to start up
RPCNFSDCOUNT=8

# Runtime priority of server (see nice(1))
RPCNFSDPRIORITY=0

# Options for rpc.mountd.
# If you have a port-based firewall, you might want to set up
# a fixed port here using the --port option. For more information,
# see rpc.mountd(8) or http://wiki.debian.org/SecuringNFS
# To disable NFSv4 on the server, specify '--no-nfs-version 4' here
RPCMOUNTDOPTS="--p 32767 --manage-gids"

# Do you want to start the svcgssd daemon? It is only required for Kerberos
# exports. Valid alternatives are "yes" and "no"; the default is "no".
KEEP_SVCSSD=""

# Options for rpc.svcgssd.
RPCSVCGSSDOPTS=""

```

Figura 9: Puerto de RPC fijado.

Ahora para aplicar los cambios y ver que se han realizado ejecutamos los siguientes comandos.

- 1 **# Reiniciamos el servicio tras indicar en el fichero de configuracion el puerto**
- 2 `sudo systemctl restart nfs-server.service`
- 3 **# Comprobamos que el estado del servicio active**
- 4 `sudo systemctl status nfs-server.service`
- 5 **# Comprobamos que el puerto de rpc que es de mountd se haya cambiado.**
- 6 `sudo rpcinfo -p localhost`

```

debian@ml-pedroamp:~$ sudo systemctl restart nfs-server.service
debian@ml-pedroamp:~$ sudo systemctl status nfs-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
   Active: active (exited) since Tue 2022-05-31 11:27:40 UTC; 9s ago
     Process: 1831 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
     Process: 1832 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
    Main PID: 1832 (code=exited, status=0/SUCCESS)
      CPU: 11ms

May 31 11:27:39 ml-pedroamp systemd[1]: Starting NFS server and services...
May 31 11:27:40 ml-pedroamp systemd[1]: Finished NFS server and services.
debian@ml-pedroamp:~$ sudo rpcinfo -p localhost
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 32767 mountd
100005 1 tcp 32767 mountd
100005 2 udp 32767 mountd
100005 2 tcp 32767 mountd
100005 3 udp 32767 mountd
100005 3 tcp 32767 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 46581 nlockmgr
100021 3 udp 46581 nlockmgr
100021 4 udp 46581 nlockmgr
100021 1 tcp 41801 nlockmgr
100021 3 tcp 41801 nlockmgr
100021 4 tcp 41801 nlockmgr
debian@ml-pedroamp:~$ sudo netstat -tulpn | grep 32767

```

Figura 10: Puerto de RPC fijado comprobación con rpcinfo.

El último puerto es el nlockmgr que necesita ser configurado como parámetro del kernel. El fichero de configuración está en la ruta `/etc/sysctl.d/nfs-static-ports.conf`.

```

debian@ml-pedroamp:~$ cat /etc/sysctl.d/nfs-static-ports.conf
fs.nfs.nlm_tcpport = 32768
fs.nfs.nlm_udpport = 32768

```

Figura 11: Puerto de nlockmgr fijado en la configuración.

```

1 # Recargamos los parametros del kernel. Para que el puerto se
2 sudo sysctl --system
3 # Reiniciamos el servicio de NFS
4 sudo systemctl restart nfs-server.service

```

```

debian@ml-pedroamp:~$ sudo sysctl --system
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
kernel.pid_max = 4194304
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /etc/sysctl.d/nfs-static-ports.conf ...
fs.nfs.nlm_tcpport = 32768
fs.nfs.nlm_udpport = 32768
* Applying /usr/lib/sysctl.d/protect-links.conf ...
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
* Applying /etc/sysctl.conf ...
debian@ml-pedroamp:~$ sudo systemctl restart nfs-server.service
debian@ml-pedroamp:~$ sudo rpcinfo -p localhost

```

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	32767	mountd
100005	1	tcp	32767	mountd
100005	2	udp	32767	mountd
100005	2	tcp	32767	mountd
100005	3	udp	32767	mountd
100005	3	tcp	32767	mountd
100003	3	tcp	2049	nfs
100003	4	tcp	2049	nfs
100227	3	tcp	2049	nfs
100003	3	udp	2049	nfs
100227	3	udp	2049	nfs
100021	1	udp	32768	nlockmgr
100021	3	udp	32768	nlockmgr
100021	4	udp	32768	nlockmgr
100021	1	tcp	32768	nlockmgr
100021	3	tcp	32768	nlockmgr
100021	4	tcp	32768	nlockmgr

```

debian@ml-pedroamp:~$

```

Figura 12: Puerto de nlockmgr fijado comprobación con rpcinfo

1.4.1. Reglas de iptables

Para las reglas de iptables solo tenemos que incluir los siguientes puertos, desde el puerto de portmap hasta el de RPC.

```
1 # Servidor de NFS
2 # portmapper — mapeador de puertos de RPC
3 iptables -A INPUT -p tcp -s ${m2} --dport 111 -m state --state NEW,
  ESTABLISHED -j ACCEPT
4 iptables -A OUTPUT -p tcp -d ${m2} --sport 111 -m state --state ESTABLISHED
  -j ACCEPT
5
6 iptables -A INPUT -p udp -s ${m2} --dport 111 -m state --state NEW,
  ESTABLISHED -j ACCEPT
7 iptables -A OUTPUT -p udp -d ${m2} --sport 111 -m state --state ESTABLISHED
  -j ACCEPT
8
9 # NFS Port
10 iptables -A INPUT -p tcp -s ${m2} --dport 2049 -m state --state NEW,
  ESTABLISHED -j ACCEPT
11 iptables -A OUTPUT -p tcp -d ${m2} --sport 2049 -m state --state
  ESTABLISHED -j ACCEPT
12
13 iptables -A INPUT -p udp -s ${m2} --dport 2049 -m state --state NEW,
  ESTABLISHED -j ACCEPT
14 iptables -A OUTPUT -p udp -d ${m2} --sport 2049 -m state --state
  ESTABLISHED -j ACCEPT
15
16 # RPC Remote Procedure Call
17 iptables -A INPUT -p tcp -s ${m2} --dport 32767 -m state --state NEW,
  ESTABLISHED -j ACCEPT
18 iptables -A OUTPUT -p tcp -d ${m2} --sport 32767 -m state --state
  ESTABLISHED -j ACCEPT
19
20 iptables -A INPUT -p udp -s ${m2} --dport 32767 -m state --state NEW,
  ESTABLISHED -j ACCEPT
21 iptables -A OUTPUT -p udp -d ${m2} --sport 32767 -m state --state
  ESTABLISHED -j ACCEPT
22
23 # nlockmgr — Network Lock Manager
24 iptables -A INPUT -p tcp -s ${m2} --dport 32768 -m state --state NEW,
  ESTABLISHED -j ACCEPT
25 iptables -A OUTPUT -p tcp -d ${m2} --sport 32768 -m state --state
  ESTABLISHED -j ACCEPT
26
27 iptables -A INPUT -p udp -s ${m2} --dport 32768 -m state --state NEW,
  ESTABLISHED -j ACCEPT
28 iptables -A OUTPUT -p udp -d ${m2} --sport 32768 -m state --state
  ESTABLISHED -j ACCEPT
```

Las reglas completas son en m1-pedroamp como servidor NFS son:

```
1  #!/bin/bash
2
3  # VARIABLES for IP
4  m2="192.168.122.119"
5  m3="192.168.122.57"
6
7  # Flushing iptables
8  iptables --flush
9
10 # White Listing Firewall
11 iptables -P INPUT DROP
12 iptables -P OUTPUT DROP
13 iptables -P FORWARD DROP
14
15 # Permitir conexiones en localhost
16 iptables -A INPUT -i lo -j ACCEPT
17 iptables -A OUTPUT -o lo -j ACCEPT
18
19 # SSH rules
20 # La cadena de INPUT siguiendo el TCP Handshake se establece un registro en
    la tabla de estados de IPTABLES (cortafuegos con estado STATEFUL)
21 # en la que toda la conexion nueva *NEW debe ser registrada en dicha tabla
    y pasa a ser como conexion
22 # establecida. Dicho registro en el estado ESTABLISHED permite la siguiente
    cadena en la cual sale una conexion de
23 # SSH permitida a traves del puerto 22.
24
25 iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j
    ACCEPT
26 iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
27
28 # ICMP echo PING RULES ALLOW
29 iptables -A INPUT -p icmp --icmp-type echo-request -m state --state NEW,
    ESTABLISHED,RELATED -j ACCEPT
30 iptables -A OUTPUT -p icmp --icmp-type echo-reply -m state --state
    ESTABLISHED,RELATED -j ACCEPT
31
32 # Regla de contacto con desde el balanceador a sus servidores
33 # Solo se permite el contacto del balanceador para el puerto HTTPs 443
34 iptables -A INPUT -p tcp -s ${m3} --dport 443 -m state --state NEW,
    ESTABLISHED -j ACCEPT
35 iptables -A OUTPUT -p tcp -d ${m3} --sport 443 -m state --state ESTABLISHED
    -j ACCEPT
36
37 # Regla para permitir el uso de un repositorio de APT
38 iptables -A OUTPUT -p tcp -d 199.232.182.132 --dport 80 -m state --state
    NEW,ESTABLISHED -j ACCEPT
39 iptables -A INPUT -p tcp -s 199.232.182.132 --sport 80 -m state --state
    ESTABLISHED -j ACCEPT
40
41 # Regla de DNS
42 iptables -A OUTPUT -p udp --dport 53 -m state --state NEW,ESTABLISHED -j
```

```

ACCEPT
43 iptables -A INPUT -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT
44
45 # Regla de MariaDB
46 # Entrada del contacto con el otro Primario Esclavo
47 iptables -A OUTPUT -p tcp -d ${m2} --dport 3306 -m state --state NEW,
    ESTABLISHED -j ACCEPT
48 iptables -A INPUT -p tcp -s ${m2} --sport 3306 -m state --state ESTABLISHED
    -j ACCEPT
49
50 # Salida para contactar con el otro Primario Esclavo
51 iptables -A INPUT -p tcp -s ${m2} --dport 3306 -m state --state NEW,
    ESTABLISHED -j ACCEPT
52 iptables -A OUTPUT -p tcp -d ${m2} --sport 3306 -m state --state
    ESTABLISHED -j ACCEPT
53
54 # Servidor de NFS
55 # portmapper - mapeador de puertos de RPC
56 iptables -A INPUT -p tcp -s ${m2} --dport 111 -m state --state NEW,
    ESTABLISHED -j ACCEPT
57 iptables -A OUTPUT -p tcp -d ${m2} --sport 111 -m state --state ESTABLISHED
    -j ACCEPT
58
59 iptables -A INPUT -p udp -s ${m2} --dport 111 -m state --state NEW,
    ESTABLISHED -j ACCEPT
60 iptables -A OUTPUT -p udp -d ${m2} --sport 111 -m state --state ESTABLISHED
    -j ACCEPT
61
62 # NFS Port
63 iptables -A INPUT -p tcp -s ${m2} --dport 2049 -m state --state NEW,
    ESTABLISHED -j ACCEPT
64 iptables -A OUTPUT -p tcp -d ${m2} --sport 2049 -m state --state
    ESTABLISHED -j ACCEPT
65
66 iptables -A INPUT -p udp -s ${m2} --dport 2049 -m state --state NEW,
    ESTABLISHED -j ACCEPT
67 iptables -A OUTPUT -p udp -d ${m2} --sport 2049 -m state --state
    ESTABLISHED -j ACCEPT
68
69 # RPC Remote Procedure Call
70 iptables -A INPUT -p tcp -s ${m2} --dport 32767 -m state --state NEW,
    ESTABLISHED -j ACCEPT
71 iptables -A OUTPUT -p tcp -d ${m2} --sport 32767 -m state --state
    ESTABLISHED -j ACCEPT
72
73 iptables -A INPUT -p udp -s ${m2} --dport 32767 -m state --state NEW,
    ESTABLISHED -j ACCEPT
74 iptables -A OUTPUT -p udp -d ${m2} --sport 32767 -m state --state
    ESTABLISHED -j ACCEPT
75
76 # nlockmgr - Network Lock Manager
77 iptables -A INPUT -p tcp -s ${m2} --dport 32768 -m state --state NEW,
    ESTABLISHED -j ACCEPT
78 iptables -A OUTPUT -p tcp -d ${m2} --sport 32768 -m state --state

```

```
ESTABLISHED -j ACCEPT
79
80 iptables -A INPUT -p udp -s ${m2} --dport 32768 -m state --state NEW,
    ESTABLISHED -j ACCEPT
81 iptables -A OUTPUT -p udp -d ${m2} --sport 32768 -m state --state
    ESTABLISHED -j ACCEPT
```

En el cliente m2-pedroamp especificamos:

```
1 # Servidor de NFS
2 # portmapper — mapeador de puertos de RPC
3 iptables -A OUTPUT -p tcp -d ${m1} --dport 111 -m state --state NEW,
  ESTABLISHED -j ACCEPT
4 iptables -A INPUT -p tcp -s ${m1} --sport 111 -m state --state ESTABLISHED
  -j ACCEPT
5
6 iptables -A OUTPUT -p udp -d ${m1} --dport 111 -m state --state NEW,
  ESTABLISHED -j ACCEPT
7 iptables -A INPUT -p udp -s ${m1} --sport 111 -m state --state ESTABLISHED
  -j ACCEPT
8
9 # NFS Port
10 iptables -A OUTPUT -p tcp -d ${m1} --dport 2049 -m state --state NEW,
  ESTABLISHED -j ACCEPT
11 iptables -A INPUT -p tcp -s ${m1} --sport 2049 -m state --state ESTABLISHED
  -j ACCEPT
12
13 iptables -A OUTPUT -p udp -d ${m1} --dport 2049 -m state --state NEW,
  ESTABLISHED -j ACCEPT
14 iptables -A INPUT -p udp -s ${m1} --sport 2049 -m state --state ESTABLISHED
  -j ACCEPT
15
16 # RPC Remote Procedure Call
17 iptables -A OUTPUT -p tcp -d ${m1} --dport 32767 -m state --state NEW,
  ESTABLISHED -j ACCEPT
18 iptables -A INPUT -p tcp -s ${m1} --sport 32767 -m state --state
  ESTABLISHED -j ACCEPT
19
20 iptables -A OUTPUT -p udp -d ${m1} --dport 32767 -m state --state NEW,
  ESTABLISHED -j ACCEPT
21 iptables -A INPUT -p udp -s ${m1} --sport 32767 -m state --state
  ESTABLISHED -j ACCEPT
22
23 # nlockmgr — Network Lock Manager
24 iptables -A OUTPUT -p tcp -d ${m1} --dport 32768 -m state --state NEW,
  ESTABLISHED -j ACCEPT
25 iptables -A INPUT -p tcp -s ${m1} --sport 32768 -m state --state
  ESTABLISHED -j ACCEPT
26
27 iptables -A OUTPUT -p udp -d ${m1} --dport 32768 -m state --state NEW,
  ESTABLISHED -j ACCEPT
28 iptables -A INPUT -p udp -s ${m1} --sport 32768 -m state --state
  ESTABLISHED -j ACCEPT
```


Las reglas completas en m2-pedroamp como cliente NFS son:

```
1 #!/bin/bash
2
3 # VARIABLES for IP
4 m2="192.168.122.119"
5 m3="192.168.122.57"
6
7 # Flushing iptables
8 iptables --flush
9
10 # White Listing Firewall
11 iptables -P INPUT DROP
12 iptables -P OUTPUT DROP
13 iptables -P FORWARD DROP
14
15 # Permitir conexiones en localhost
16 iptables -A INPUT -i lo -j ACCEPT
17 iptables -A OUTPUT -o lo -j ACCEPT
18
19 # SSH rules
20 # La cadena de INPUT siguiendo el TCP Handshake se establece un registro en
    la tabla de estados de IPTABLES (cortafuegos con estado STATEFUL)
21 # en la que toda la conexion nueva *NEW debe ser registrada en dicha tabla
    y pasa a ser como conexion
22 # establecida. Dicho registro en el estado ESTABLISHED permite la siguiente
    cadena en la cual sale una conexion de
23 # SSH permitida a traves del puerto 22.
24
25 iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j
    ACCEPT
26 iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
27
28 # ICMP echo PING RULES ALLOW
29 iptables -A INPUT -p icmp --icmp-type echo-request -m state --state NEW,
    ESTABLISHED,RELATED -j ACCEPT
30 iptables -A OUTPUT -p icmp --icmp-type echo-reply -m state --state
    ESTABLISHED,RELATED -j ACCEPT
31
32 # Regla de contacto con desde el balanceador a sus servidores
33 # Solo se permite el contacto del balanceador para el puerto HTTPs 443
34 iptables -A INPUT -p tcp -s ${m3} --dport 443 -m state --state NEW,
    ESTABLISHED -j ACCEPT
35 iptables -A OUTPUT -p tcp -d ${m3} --sport 443 -m state --state ESTABLISHED
    -j ACCEPT
36
37 # Regla para permitir el uso de un repositorio de APT
38 iptables -A OUTPUT -p tcp -d 199.232.182.132 --dport 80 -m state --state
    NEW,ESTABLISHED -j ACCEPT
39 iptables -A INPUT -p tcp -s 199.232.182.132 --sport 80 -m state --state
    ESTABLISHED -j ACCEPT
40
41 # Regla de DNS
42 iptables -A OUTPUT -p udp --dport 53 -m state --state NEW,ESTABLISHED -j
```

```

ACCEPT
43 iptables -A INPUT -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT
44
45 # Regla de MariaDB
46 # Entrada del contacto con el otro Primario Esclavo
47 iptables -A OUTPUT -p tcp -d ${m2} --dport 3306 -m state --state NEW,
    ESTABLISHED -j ACCEPT
48 iptables -A INPUT -p tcp -s ${m2} --sport 3306 -m state --state ESTABLISHED
    -j ACCEPT
49
50 # Salida para contactar con el otro Primario Esclavo
51 iptables -A INPUT -p tcp -s ${m2} --dport 3306 -m state --state NEW,
    ESTABLISHED -j ACCEPT
52 iptables -A OUTPUT -p tcp -d ${m2} --sport 3306 -m state --state
    ESTABLISHED -j ACCEPT
53
54 # Servidor de NFS
55 # portmapper - mapeador de puertos de RPC
56 iptables -A INPUT -p tcp -s ${m2} --dport 111 -m state --state NEW,
    ESTABLISHED -j ACCEPT
57 iptables -A OUTPUT -p tcp -d ${m2} --sport 111 -m state --state ESTABLISHED
    -j ACCEPT
58
59 iptables -A INPUT -p udp -s ${m2} --dport 111 -m state --state NEW,
    ESTABLISHED -j ACCEPT
60 iptables -A OUTPUT -p udp -d ${m2} --sport 111 -m state --state ESTABLISHED
    -j ACCEPT
61
62 # NFS Port
63 iptables -A INPUT -p tcp -s ${m2} --dport 2049 -m state --state NEW,
    ESTABLISHED -j ACCEPT
64 iptables -A OUTPUT -p tcp -d ${m2} --sport 2049 -m state --state
    ESTABLISHED -j ACCEPT
65
66 iptables -A INPUT -p udp -s ${m2} --dport 2049 -m state --state NEW,
    ESTABLISHED -j ACCEPT
67 iptables -A OUTPUT -p udp -d ${m2} --sport 2049 -m state --state
    ESTABLISHED -j ACCEPT
68
69 # RPC Remote Procedure Call
70 iptables -A INPUT -p tcp -s ${m2} --dport 32767 -m state --state NEW,
    ESTABLISHED -j ACCEPT
71 iptables -A OUTPUT -p tcp -d ${m2} --sport 32767 -m state --state
    ESTABLISHED -j ACCEPT
72
73 iptables -A INPUT -p udp -s ${m2} --dport 32767 -m state --state NEW,
    ESTABLISHED -j ACCEPT
74 iptables -A OUTPUT -p udp -d ${m2} --sport 32767 -m state --state
    ESTABLISHED -j ACCEPT
75
76 # nlockmgr - Network Lock Manager
77 iptables -A INPUT -p tcp -s ${m2} --dport 32768 -m state --state NEW,
    ESTABLISHED -j ACCEPT
78 iptables -A OUTPUT -p tcp -d ${m2} --sport 32768 -m state --state

```

```
ESTABLISHED -j ACCEPT
79
80 iptables -A INPUT -p udp -s ${m2} --dport 32768 -m state --state NEW,
    ESTABLISHED -j ACCEPT
81 iptables -A OUTPUT -p udp -d ${m2} --sport 32768 -m state --state
    ESTABLISHED -j ACCEPT
```

[illegible]

Figura 13: Pruebas realizadas con iptables funcionando:

2. Bibliografía

- ¹ <https://wiki.debian.org/NFSServerSetup>
- ² <https://wiki.debian.org/SecuringNFS>
- ³ <http://recursostic.educacion.es/observatorio/web/gl/software/software-general/733-nfs-sistema-de-archivos-de-red>
- ⁴ <http://nfs.sourceforge.net/>
- ⁵ <https://man7.org/linux/man-pages/man8/mount.8.html>
- ⁶ <https://linuxide.com/understanding-each-entry-of-linux-fstab-etcfstab-file/>

¹ Configuración de Servidor NFS - Debian

² Configuración de firewall y seguridad de NFS - Debian

³ Configuración de Servidor NFS - Ajustes del archivo de exports

⁴ NFS - Explicación de algunas configuraciones del sistema.

⁵ Manpage de mount.

⁶ Opciones de mount y fstab.