

TRABAJO ASO

Lois Briones Oliveira
Brais Fernández Reyes
Fernando Losada Pérez

April 13, 2025

Contents

1	Que es NIS	5
2	NIS Arquitectura	5
2.1	Servidor Maestro NIS (NIS Master Server)	5
2.2	Servidor esclavos NIS (NIS Slave Server)	5
2.3	Cliente NIS (NIS Clients)	5
3	Dominios NIS	5
4	Como trabajar y que son los mapas en NIS	6
4.1	Como obtener información de los mapas	7
5	Comandos comunes en NIS	7
6	Netgroup en NIS	7
7	Configuración de NIS en Ubuntu Server	8
7.1	Configuración del Server	8
7.1.1	Instalación de los paquetes necesarios	8
7.1.2	Configurar el rol de servidor maestro	8
7.1.3	Configurar del fichero securenets	8
7.1.4	Iniciar los servicios NIS	8
7.1.5	Contruir los mapas de NIS	8
7.2	Configuración del Slave	9
7.2.1	Instalación y configuración del dominio	9
7.2.2	Configurar la resolución y acceso	9
7.2.3	Iniciar el slave	9
7.3	Configuración del Cliente	9
7.3.1	Instalación de paquetes	9
7.3.2	Configurar /etc/yp.conf	9
7.3.3	Configurar la búsqueda en el fichero nsswitch.conf	9
7.3.4	Inicar el servicio en el cliente	10
8	Configuración en FreeBSD	10

8.1	Conocimientos previos	10
8.2	Configuración del Server	10
8.2.1	Configuración del fichero <code>/etc/rc.conf</code>	10
8.2.2	Activación del dominio NIS	11
8.2.3	Generar los mapas en el servidor NIS	11
8.3	Configuración del Slave	12
8.3.1	Lanzar el comando de configuración	12
8.3.2	(OPCIONAL) Configuración de <code>cron</code> para mantener mapas actualizados	12
8.4	Configuración del Cliente	13
8.4.1	Configuración del fichero <code>/etc/rc.conf</code>	13
8.4.2	Creación del fichero <code>/etc/yp.conf</code>	13
9	Configuración en Solaris	14
9.1	Conocimientos previos	14
9.1.1	Conocimiento de procesos	14
9.2	Configuraciones previas a la instalación	15
9.2.1	Configuración del fichero <code>nsswitch</code>	15
9.2.2	Configuración y establecimiento del dominio NIS	15
9.2.3	Configurar el fichero <code>/etc/init/hosts</code>	16
9.3	Configuración del Server	17
9.3.1	Preparación previa de los ficheros	17
9.3.2	Preperación y configuración del Makefile	18
9.3.3	Lanzar el comando de comando de configuración: <code>ypinit</code>	19
9.3.4	(OPCIONAL) Configurar el fichero <code>securenets</code>	20
9.4	Configuración del Slave	22
9.4.1	Configurar primero el slave como cliente	22
9.4.2	Descargar paquete de configuración	22
9.4.3	Iniciar la maquina como un slave server	22
9.5	Configuración del Cliente	23
9.5.1	Problemas conexion cliente con el servidor	24
10	Que es NFS	26
10.1	Como funciona	26

11 Como configurar NFS en Ubuntu Server	26
11.1 Configuración del servidor	26
11.1.1 Instalación de paquete necesarios	26
11.1.2 Configuración de los directorios exportados	26
11.2 Configuración del cliente	27
11.2.1 Instalación de paquetes necesarios	27
11.2.2 Montaje del directorio manualmente	27
11.2.3 Montaje automatico mediante <code>/etc/fstab</code>	28
12 Como configurar NFS en FreeBSD	28
12.1 Configuración del servidor	28
12.2 Configuración del cliente	29
13 Como configurar NFS en Solaris	29
13.1 Configuración del server	29
13.2 Configuración del cliente	29
14 Inteconexión NIS y NFS	30
15 Comparación entre NIS,NIS+ y LDAP	30
15.1 Seguridad	31
15.2 Escalabilidad y arquitectura	31
15.3 Facilidad de admistración	32
15.4 Soporte actual y estado de la tecnología	32
15.5 Migración de NIS a LDAP	32

1 Que es NIS

NIS (Network Information System) es un sistema cliente-servidor creado por Sun Microsystems para centralizar la configuración de sistemas UNIX y similares, lo que permite que exista una completa compatibilidad entre ellos. Aunque originalmente se llamaba Yellow Pages (YP), tuvo que cambiar de nombre por razones legales. Aun así, muchos comandos siguen comenzando con el prefijo yp, como ypserv o ypbind. NIS permite compartir información como usuarios, contraseñas y nombres de máquinas entre varios equipos conectados a una red, haciendo que todos parezcan un solo sistema. Esto es útil especialmente en redes locales pequeñas donde configurar DNS sería innecesario o complicado. NIS está basado en RPC (Remote Procedure Call) y utiliza una estructura compuesta por servidores y clientes.

2 NIS Arquitectura

2.1 Servidor Maestro NIS (NIS Master Server)

Es el servidor principal, donde se guardan los archivos originales de configuración (como passwd, group, etc.). Es el punto central desde donde se distribuye la información a los demás equipos.

2.2 Servidor esclavos NIS (NIS Slave Server)

Mantienen copias actualizadas de los datos del servidor maestro. Sirven como respaldo en caso de fallo del maestro y ayudan a repartir la carga de trabajo, ya que los clientes se conectan al servidor que responda más rápido.

2.3 Cliente NIS (NIS Clients)

Son los equipos que consultan la información en los servidores NIS en lugar de usar archivos locales. Por ejemplo, al iniciar sesión, un cliente puede verificar usuarios y contraseñas directamente contra el servidor NIS.

3 Dominios NIS

Un dominio NIS es un conjunto lógico de máquinas que comparten la misma información de red a través de NIS. Todos los clientes y servidores deben pertenecer al mismo dominio para comunicarse correctamente.

Todo dominio NIS debe tener un nombre, este puede ser uno inventado que no sigue unas reglas concretas. Aunque sí que se recomienda que sea algo que tenga sentido si estamos configurando el servidor NIS dentro de una organización. Destacar que los nombres de dominio son sensibles a mayúsculas y minúsculas.

Un dominio NIS aunque se pueda parecer a un dominio DNS, la realidad es que no son necesariamente lo mismo. Normalmente como se comentaba arriba, los dominios NIS se suelen definir en base a la distribución administrativa de las subredes. En cambio los nombres y dominios DNS se definen según los estándares y jerarquías de nombres. Es cierto que se podría configurar de forma que coincidieran y de hecho se podría configurar NIS para que utilizara DNS para la resolución de los nombres. Pero normalmente el nombre del dominio para ambos servicios se controla de forma separada y por distintos servicios. Lo más aconsejable es que los dominios NIS sigan un orden parecido al que siguen las subredes de la empresa.

Cualquier máquina puede pertenecer a un dominio NIS, siempre que existe un servidor master para ese dominio que pueda distribuir los mapas. Destacar que solo puede haber un servidor master por dominio y tanto servidores slaves como se quiera.

4 Como trabajar y que son los mapas en NIS

El servicio NIS lo que hace para poder ofrecer los ficheros de configuración y toda la información que tiene disponible es utilizar lo que se conoce como mapas. Esto no son más que ficheros que genera apartir de la información que hay almacenada en los ficheros de configuración.

El formato de los mapas depende del sistema operativo, en el caso de Solaris o de Ubuntu Server siguen manteniendo el formato que tenían los antiguos ficheros en todos los sistemas operativos que implementaban NIS. En este caso sería *nbdm*, en cambio, en los sistema como FreeBSD, el formato de los mapas es distinto esto es debido a la licencia que se impuso sobre el formato. Algunos desaroyadores pagaron para así manter el formato pero en el caso de FreeBSD. Razón por la cual utiliza *Berkeley DB hash method*, que funciona un poco distinto, en vez de tener varios mapas separados como era el funcionamiento antiguo donde se tenía un mapa dependiendo de la clave de busqueda que se fuera a utilizar, en el caso de FreeBSD todo se almacena en el mismo fichero. Esto no va a dificultar la comunicación simplemente el servicio **ypserv** tiene que saber el formato y podrá trabajar con ambos.

Aunque hay está pequeña diferencia al final la esencia de los mapas NIS es una tabla hash donde se tienen los datos almacenados en un ficheros y indexados por una clave que funciona para encontrar la información que buscamos de forma rápida. La idea es que la clave que se utilice varie en función de la información que buscamos. De forma que aunque los ficheros que se generan varian, el funcionamiento por debajo sea el mismo de manera que para un mismo fichero pueden existir varios mapas cada uno de ellos ordenado por una clave apartir de la cual luego queremos buscar la información. Por ejemplo, supongamos que tenemos el fichero `/etc/hosts` pues para este podemos tener un mapa que podría ser `hosts.byaddr` que nos permite apatir de una ip obtener el nombre del host o por lo contrario podríamos tener `hosts.byname`, que apartir del nombre del host nos permite obtener la ip.

Estos mapas están alamacenados en el master server aunque también pueden haber copias en lo que serían replicas de este, conocidas como slave server.

Como se explicaba arriba cuando se establece el servicio NIS en la red este trabajo sobre un dominio, para cada uno de ellos se pueden generar mapas independientes.

Todos los mapas se generar o modificar apartir de un fichero que se suele llamar Makefile. Aparte normalmente todos los mapas que se generan siguen el mismo sistema de nombres: *nombre del fichero original + clave de busqueda*. Aparte de los ficheros comunes también se pueden añadir a NIS otros mapas que son creados por necesidad o que se añaden cuando se instala una aplicación nueva.

Por norma general cuando se generan los mapas estos se almacenan normalmente en `/var/yp` + el dominio NIS que tengamos configurado. Quedando que la ruta a los mapas es `/var/yp/domainname`. Destacar además que para cada dominio NIS puedes tener distintos mapas.

Aparte también para que sea más sencillo hacer referencia a los mapas NIS asignando a un mapa un alias. Normalmente estes se guardan en `/var/yp/nicknames` y no es más que un fichero que contiene distintas entradas donde cada una, lo que tiene es, primero el nickname y acontinuación separado por espacio el nombre completo del mapa. Está lista puede ser modificada o actualiza. Aparte también existe un limite que establece que solo puede haber 500 nicknames.

Para obtener información acerca de los nicknames que hay disponibles para cada mapa hay dos posibles comandos que nos devuelven está información:

- `ypcat -x`
- `ypwhich -x`

Destacar que los mapas tienen un problema bastante importante y es que no hay ningún metodo de autenticación por tanto cualquiera que logre alcanzar el servidor va a poder conectarse y aparte obtener información ya que no está encriptada ni controlada.

También es importante no olvidar que siempre que se modifique un mapa va a ser necesario regenerarlo utilizando

el **Makefile**. Además en caso de tener servidores slave configuradores tendríamos que utilizar el comando **yppush** para actualizar los mapas que tienen guardados.

4.1 Como obtener información de los mapas

- Para listar toda la información que almacena un mapa, pero no las claves
`ypcat mapname`
- Para obtener toda la información que almacena un mapa, tanto información como claves
`ypcat -k mapname`
- Para obtener todos los mapas disponibles y el master que los tiene
`ypwhich -m`
- Para obtener el master para un mapa en particular
`ypwhich -m mapname`

5 Comandos comunes en NIS

Table 1: Resumen de comandos NIS

Comando	Descripción
ypcat	Sirve para obtener el contenido de los mapas NIS.
ypinit	Comando que se utiliza para preparar el master, el cliente o el slave. Dependiendo del flag con que se ejecute, genera los mapas y levanta los servicios en el master, o configura los ficheros necesarios y servicios en el cliente o el slave.
ypmatch	Imprime el valor de las claves de un mapa NIS.
yppoll	Muestra qué versión del mapa NIS está corriendo en el servidor. También devuelve el servidor maestro del mapa.
yppush	Copia la nueva versión del mapa NIS desde el servidor maestro a los esclavos. Se debe ejecutar desde el master server.
ypset	Hace que un proceso de 'ypbind' se vincule a un servidor indicado. No se recomienda su uso habitual por razones de seguridad.
ypwhich	Muestra qué servidor NIS está usando el cliente para solicitar información.
ypxfr	Extrae un mapa NIS desde un servidor remoto al directorio local '/var/yp/domain', utilizando NIS como medio de transporte. Puede ejecutarse manualmente o desde 'crontab', y también lo llama 'ypserv' para iniciar una transferencia.

6 Netgroup en NIS

Para simplificar la tarea de administración NIS no ofrece la capacidad de trabajar con netgroups, que no son más que una terna de usuarios, maquinas y dominios, a los cuales, se les puede asociar un nombre. Destacar que no asignan permisos directamente si no que sus nombres se usan dentro de los ficheros de configuración para simplificar el control de acceso.

La sintaxis y la configuración puede variar dependiendo del sistema operativo que se use para estar seguro se recomienda consultar **man netgroup**. Obviamente también el fichero de configuración donde hay que añadir las entradas puede variar.

Aunque la sintaxis es bastante genérica y seguiría un formato parecido al siguiente:

nombre-grupo (host,user,domain)

7 Configuración de NIS en Ubuntu Server

7.1 Configuración del Server

7.1.1 Instalación de los paquetes necesarios

En Ubuntu, es necesario instalar el paquete `nis` (que incluye el servicio NIS y las utilidades) junto con el servicio RPC (`rpcbind`). Para ello tendríamos que:

```
sudo apt update && sudo apt install nis rpcbind
```

Durante la instalación, se le pedirá que indique el nombre de dominio NIS. Sería necesario tener ya pensado un nombre de dominio NIS para su red. Si la instalación inicial se realiza antes de configurar el servidor, es normal que `ypbind` falle temporalmente mientras no haya un servidor NIS disponible.

7.1.2 Configurar el rol de servidor maestro

Edite el archivo `/etc/default/nis` y asegúrese de que la línea `NISMASTER` esté establecida como `NISMASTER=master`. Esto indica que este sistema actuará como servidor maestro NIS.

7.1.3 Configurar del fichero `securenets`

El servicio NIS utiliza RPC, por lo que conviene restringir qué redes pueden acceder a los mapas. Para ello habría que editar el archivo `/etc/ypserv.securenets` para especificar las subredes autorizadas. De forma predeterminada, este archivo suele permitir acceso global con una entrada `0.0.0.0 0.0.0.0`, habría que eliminarla y añadir las redes de su organización, limitando las consultas NIS.

7.1.4 Iniciar los servicios NIS

Hay que asegurarse los servicios RPC y NIS se están ejecutando. En caso de que estén iniciados sería interesante reiniciarlos para cargar la nueva configuración.

```
sudo systemctl restart rpcbind
```

7.1.5 Contruir los mapas de NIS

Para crear los mapas en NIS habría que ejecutar el siguiente comando:

```
sudo /usr/lib/yp/ypinit -m
```

Durante la ejecución se va a lanzar un proceso de configuración, que va a pedir una lista de servidores NIS, aquí habría que ingresar el nombre del servidor maestro. Y después habría que añadir los nombres de los servidores slave que quisiéramos tener. Este proceso lo que va a hacer es ejecutar `/var/yp/Makefile` que apartir de los ficheros de configuración locales (`/etc/passwd`, `/etc/group`,...) va a crearlos mapas.

Una vez que terminemos para confirmar que se crearon los mapas podríamos mirar `/var/yp/<nombre_dominio>`.

7.2 Configuración del Slave

7.2.1 Instalación y configuración del dominio

En cada servidor esclavo, instale los mismos paquetes (nis, rpcbind) y asegúrese de configurar el dominio NIS durante la instalación con el mismo nombre de dominio definido en el maestro.

7.2.2 Configurar la resolución y acceso

Habría que configurar en el maestro y en los slave el fichero `/etc/hosts` con las IP del maestro y de los esclavos, asociandolos con nombres. También será necesario modificar `/etc/ypserv.securenets` para permitir la red interna de forma coherente con el maestro

7.2.3 Iniciar el slave

En el servidor esclavo habría que ejecutar:

```
sudo /usr/lib/yp/ypinit -s <host_maestro> #Se podría usar el hostname o la IP
```

Este comando lo que va a hacer es solicitar al maestro las copias de los mapas y configurará el esclavo. Trás completarse, el esclavo tendrá los mapas en `/var/yp/<dominio>` y comenzará a atender consultas NIS para los clientes.

7.3 Configuración del Cliente

7.3.1 Instalación de paquetes

Habría que instalar el paquete nis (rpcbind también si no está instalado). Durante la instalación se pedirá introducir el nombre del dominio NIS donde queremos conectar el cliente. No se quiere instalar `ypserv` en los clientes, solo las utilidades y el cliente `ypbind`

7.3.2 Configurar `/etc/yp.conf`

Habría que editar o crear si no existe el fichero `/etc/yp.conf` para indicar el dominio NIS y el servidor a usar. Habría que agregar una linea que tuviera la siguiente sintaxis:

```
domain <nombre_dominio> server <ip_servidor_nis> #Se podría poner el hostname de la maquina
```

De está manera le indicariamos al cliente que servidor NIS tiene que contactar para un dominio determinado.

7.3.3 Configurar la busqueda en el fichero `nsswitch.conf`

Habría que editar el fichero `/etc/nsswitch.conf` para incluir a que se utilize NIS para consultar la configuración. Se podría hacer para todos los ficheros que están especificados aquí y simplemente habría que añadir `nis`, un ejemplo podría ser:

```
passwd: files nis
hosts: files nis
```

De esta forma estaríamos indicándole al sistema que primero debe consultar primero los archivos locales y luego NIS si no se encuentra la información en local.

7.3.4 Inicar el servicio en el cliente

Habría que activar o reiniciar ypbind, dependiendo de si estuviera encendido o no.

- El servicio está apagado

```
sudo systemctl enable rpcbind nis
```

- El servicio está encendido

```
sudo systemctl restart rpcbind nis
```

8 Configuración en FreeBSD

8.1 Conocimientos previos

En FreeBSD, existen 4 servicios importantes para el uso de NIS:

- **rpcbind:** Habilita las llamadas a procedimientos remotos (RPC), esenciales para que funcione tanto un cliente como un servidor NIS.
- **ypbind:** Es el servicio que conecta un cliente NIS con su servidor. Usa el nombre del dominio NIS y RPC para establecer la conexión. Si este servicio no está en ejecución en el cliente, no podrá acceder a los datos NIS.
- **ypserv:** Es el proceso principal del servidor NIS. Si se detiene, el servidor no podrá responder a las solicitudes. Por eso se recomienda tener un servidor esclavo para respaldo.
- **rpc.yppasswdd:** Solo se ejecuta en el servidor maestro NIS. Permite que los usuarios cambien sus contraseñas desde cualquier cliente. Si no está activo, los usuarios tendrán que iniciar sesión directamente en el servidor maestro para cambiar su contraseña.

8.2 Configuración del Server

8.2.1 Configuración del fichero /etc/rc.conf

En el archivo /etc/rc.conf se definen variables que indican qué servicios deben arrancar con el sistema.

Aquí tendremos que añadir las siguientes líneas:

- nisdomianname="midominio", habría que definir le nombre del dominio NIS
- rpcbind_enable="YES", activa el servicio RPC, necesario para que NIS funcione
- nis_server_enable="YES", habilita el servidor NIS
- nis_yppasswd_enable="YES", permite que los usuarios cambien su contraseña desde el cliente NIS

```
$ cat /etc/rc.conf | grep -E 'server|rpc|yppasswd|dominio
nisdomainname="midominio"
rpcbind_enable="YES"
nis_yppasswdd_enable="YES"
nis_server_enable="YES"
nfs_server_enable="YES"
```

Figure 1: Fichero de configuración rc.conf en Master NIS FreeBSD

8.2.2 Activación del dominio NIS

El comando `domainname` establece temporalmente el dominio NIS para la sesión actual(hasta el próximo reinicio).

En caso de querer guardar el nombre del dominio de forma permanente para que se cargue automáticamente al reiniciar tendríamos que realizar lo siguiente:

```
echo "midominio" | sudo tee /etc/defaultdomain
```

Para aplicar ambas configuraciones tenemos 2 opciones:

- Reiniciar la maquina para que se activen los servicios
- Reiniciar a mano los servicios:

```
service <nombreservicio> start
```

8.2.3 Generar los mapas en el servidor NIS

Los mapas NIS se generan a partir de los ficheros de configuración que se encuentran en el directorio `/etc`. Por razones de seguridad (para que no se puedan propagar las contraseñas por todos los servidores del dominio NIS), el archivo *master.passwd* tenemos que copiarlo nosotros manualmente en el directorio `/var/yp`, generándose el mapa a partir de esa copia del archivo.

```
cp /etc/master.passwd /var/yp/master.passwd
```

Ahora se puede generar los mapas con la ayuda del script que tiene el comando `ypinit`. El cual va a utilizar el fichero *Makefile* que se encuentra en el directorio `/var/yp`. Por tanto para iniciar todo el proceso tendríamos que realizar lo siguiente:

```
ypinit -m
```

Una vez que se inicie el proceso nos pedirá el dominio donde queremos configurar el server NIS. Aparte también nos va a pedir que introduzcamos los hosts, en este caso tendríamos que poner el del propio servidor. Aunque se podrían añadir otros servidores que harían del slave del master.

```
# ypinit -m
Server Type: MASTER Domain: midominio

Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n]

Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.

Can we destroy the existing /var/yp/midominio and its contents? [y/n: n] y

At this point, we have to construct a list of this domains YP servers.
aso3 is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
    master server : aso3
    next host to add: ^D
The current list of NIS servers looks like this:

aso3

Is this correct? [y/n: y] y
```

Figure 2: Salida del comando ypinit -m al ser lanzado en server FreeBSD

8.3 Configuración del Slave

OJO: Antes de continuar con la configuración es necesario que en el **MASTER** configuremos un par de cosas. tenemos que cambiar la línea *NOPUSH* del archivo */var/yp/makefile*, para que diga *NOPUSH = "False"*. Si no lo hacemos, no enviará los mapas a los servidores esclavos. Eso significa que los esclavos tendrán mapas desactualizados y los clientes que se conecten a ellos podrían recibir datos incorrectos o antiguos.

8.3.1 Lanzar el comando de configuración

Vamos a tener que volver a utilizar el mismo comando que en el master:

```
ypinit -s
```

En este caso el proceso va a ser muy similar, en primer lugar te va a solicitar el dominio donde queremos configurar el slave y después tendremos que introducir el nombre de la maquina que hace de servidor maestro.

8.3.2 (OPCIONAL) Configuración de cron para mantener mapas actualizados

20	*	*	*	*	root	/usr/libexec/ypxfr passwd.byname
21	*	*	*	*	root	/usr/libexec/ypxfr passwd.byuid

Figure 3: Salida fichero cron para configurar ypxfr para controlar los mapas

Si quisiéramos, podríamos añadir entradas de este estilo al cron para forzar las actualizaciones de los mapas del servidor maestro en el servidor esclavo, aunque realmente el maestro ya intentará realizar las actualizaciones automáticamente.

8.4 Configuración del Cliente

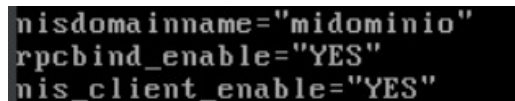
8.4.1 Configuración del fichero `/etc/rc.conf`

Tendríamos que modificar tres líneas. Podría pasar que las dos primeras podrían encontrarse ya introducidas en el fichero.

```
nisdomainname="midominio"  
rpcbind_enable="YES"  
nis_client_enable="YES"
```

En este caso `nisdomainname` debería de ser el mismo que pusimos en el mismo dominio en el que se encuentra el Master y el Slave.

En el caso de `nis_client_enable` hace que se active `ypbind` al iniciar la maquina, este va a ser el demonio que permite que el servidor se conecte con el servidor NIS.



```
nisdomainname="midominio"  
rpcbind_enable="YES"  
nis_client_enable="YES"
```

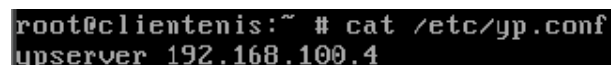
Figure 4: Fichero configuración `rc.conf` en el cliente NIS

8.4.2 Creación del fichero `/etc/yp.conf`

En este caso va a ser un fichero que vamos a tener que crear en caso de que no este ya creado. Y aquí va a ser donde especificaremos el servidor al que nos queremos conectar.

La sintaxis de fichero sería la siguiente:

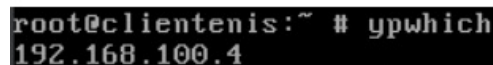
```
ypserver <ip-master>
```



```
root@clientenis:~ # cat /etc/yp.conf  
ypserver 192.168.100.4
```

Figure 5: Salida de ejemplo del fichero `yp.conf` en el cliente de NIS en FreeBSD

Después de esto ya tendríamos el cliente configurado, si quisiéramos comprobar si tenemos la configuración bien hecha podríamos utilizar `ypwhich`. Si todo estuviera bien configurado debería de devolver la ip del servidor NIS al que estamos conectados.



```
root@clientenis:~ # ypwhich  
192.168.100.4
```

Figure 6: Salida del comando `ypwhich` en el cliente una vez configurado en FreeBSD

Otra opción válida sería utilizar el comando `ypcat` para pedir un fichero al servidor.

```
$ ypcat passwd
smmsp:*:25:25:Sendmail Submission User:/var/spool/clientmqueue:/usr/sbin/nologin
hast:*:845:845:HAST unprivileged user:/var/empty:/usr/sbin/nologin
usuario:*:1001:1001:user25:/home/usuario:/bin/sh
bin:*:3:7:Binaries Commands and Source:/usr/sbin/nologin
tty:*:4:65533:Tty Sandbox:/usr/sbin/nologin
ypldap:*:160:160:YP LDAP unprivileged user:/var/empty:/usr/sbin/nologin
avahi:*:558:558:Avahi Daemon User:/nonexistent:/usr/sbin/nologin
kmem:*:5:65533:KMem Sandbox:/usr/sbin/nologin
sshd:*:22:22:Secure Shell Daemon:/var/empty:/usr/sbin/nologin
proxy:*:62:62:Packet Filter pseudo-user:/nonexistent:/usr/sbin/nologin
unbound:*:59:59:Unbound DNS Resolver:/var/unbound:/usr/sbin/nologin
pflogd:*:64:64:pflogd privsep user:/var/empty:/usr/sbin/nologin
tests:*:977:977:Unprivileged user for tests:/nonexistent:/usr/sbin/nologin
nobody:*:65534:65534:Unprivileged user:/nonexistent:/usr/sbin/nologin
cups:*:193:193:Cups Owner:/nonexistent:/usr/sbin/nologin
colord:*:970:970:colord color management daemon:/nonexistent:/usr/sbin/nologin
root:*:0:0:Charlie &:/root:/bin/sh
toor:*:0:0:Bourne-again Superuser:/root:
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5:System &:/usr/sbin/nologin
man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin
games:*:7:13:Games pseudo-user:/usr/sbin/nologin
news:*:8:8:News Subsystem:/usr/sbin/nologin
```

Figure 7: Salida del comando ypcat una vez configurado el cliente en FreeBSD

9 Configuración en Solaris

9.1 Conocimientos previos

9.1.1 Conocimiento de procesos

La configuración se va a realizar sobre un Solaris 11.4, en esta versión el utiliza SMF (Service Management Facility) para gestionar los servicios del sistema. SMF se introdujo con el objetivo de reemplazar a los antiguos scripts de inicio (/etc/init.d) y proporcionar así una forma más robuste y controlada de administrar los servicios.

Es algo un poco complejo, no sé pretende explicar todo en profundidad para si dar unas nociones basicas para que así se mucho más comodo seguir la explicación.

Tanto los servicios como las instancias de los servicios se nombran utilizando una cadenas de caracteres, por ejemplo, para el servicio que implementa **cron** tiene el siguiente nombre: **system/cron** y luego por cada instancia que se genera del servicio se indica con un nombre por defecto el que se usa, aunque suene revundante, **default**.

Aunque es cierto que el funcionamiento es como se explica arriba la mayoría de herramientas se refieren a las instancias de los servicios utilizando el FMRI (fault management resource identifiers), el cual combina, el nombre del servicio y el nombre de la instancia. Para el ejemplo anterior el FMRI para la instancia de cron sería **svc:/system/cron:default**.

Para poder trabajar con los servicios tenemos varias herramientas que nos lo permiten aunque las principales y las que se van a utilizar en la explicación son:

- **svcs**, el siguiente comando nos muestra información sobre las instancias de un servicio. Para más información consultar (man svcs)
- **svcadm**, el siguiente comando permite administrar los servicios que se ejecutan dentro de SMF. Para más información consultar **man svcadm**
- **svccfg**, el siguiente comando nos permite modificar la configuración que hay almacenada de los servicios. Normalmente hasta que no se reinicia el servicio no suele tener efecto. Para más información consultar **man svccfg**

Finalmente destacar que los servicios pueden tener diferentes estados y es importante entenderlos también.

- disabled: La implementación del servicio aun no ha arrancado o ha sido parada
- offline: El servicio no está corriendo, pero en caso de que se cumplan las dependencias iniciará
- online: El servicio inicio correctamente
- degraded: El servicio está corriendo pero con una reducción en sus funcionalidades o eficiencia
- maintenance: El servicio no puedo arrancar debido algún error y se requiere una intervención
- uninitialized: El servicio fue registrado en el sistema pero todavía no se inicio por el servicio reiniciador(`svc.startd`, para más información consultar `man svc.startd`)

9.2 Configuraciones previas a la instalación

9.2.1 Configuración del fichero nsswitch

Antes de empezar habría que realizar una configuración tanto en el cliente como en el servidor. Para poder asegurar que el sistema operativo utilice el servidor NIS cuando busque información acerca de la configuración local. Para esto habría que modificar el fichero `nsswitch.conf`. El problema es que en solaris no se puede modificar manualmente accediendo al fichero y añadiendo las entradas a mano. El fichero se mantiene por comodidad, pero para poder cambiar el funcionamiento hay que utilizar un servicio que hay disponible

```
svccfg -s system/name-service/switch
svc:/system/name-service/switch> setprop config/default = "files nis"
svc:/system/name-service/switch> quit

svcadm refresh name-service/switch
```

9.2.2 Configuración y establecimiento del dominio NIS

Aparte de configurar el fichero `nsswitch`, también es necesario determinar cual va a ser el dominio dentro del cual va a trabajar nuestra maquina.

Para ello habría que decidir un nombre que asociarle. Hay que destacar que este dominio no tiene nada que ver con un dominio web y tampoco es algo que necesitemos que resuelva el DNS. Es algo interno para organizar los servidores NIS. Podemos decidir que maquinas pertenecen a cada dominio de forma independiente. Destacar también que una misma maquina puede pertenecer a varios dominio a la vez no necesariamente tiene que trabajar solamente dentro de uno. Pero eso es algo que veremos más adelante.

Para poder asinar un dominio a la maquina será necesario, hacerlo desde un servicio que nos ofrece la posibilidad de modificarlo igual que para el fichero `nsswitch`. Sería lo siguiente:

```
svccfg -s nis/domain
svc:/network/nis/domain>setprop config/domainname = "nombre"
svc:/network/nis/domain>quit
svccfg -s nis/domain:default refresh
svcadm enable nis/domain
```

Lo que pongamos como nombre del dominio no tiene que seguir un fomato especifico, aunque si es cierto que si lo estamos configurando dentro de una empresa sería interesante, que fuera por ejemplo si estamos configurado un nis para varios departamentos diferenciar un dominio para cada departamento entonces sería interesante que el nombre fuera autoexplicativo como por ejemplo: "salesdomain.nowhere.org", aunque si ponemos algo como

Con esto lo que tendríamos sería el servicio configurado y funcionando ahora si quisiéramos consultar el dominio que tiene establecido nuestra máquina podríamos hacerlo utilizando el siguiente comando:

Otra posible opción sería mirar el contenido que hay en el fichero `/etc/defaultdomain`, aunque esto es lo que devuelve el comando `domainname`

Para asegurarnos que tanto el cliente como el server saben como resolver el nombre que le tenemos que indicar, es necesario modificar el fichero `/etc/init/hosts`. En el caso del servidor puede parecer que no tenga mucho sentido pero es una opción que se recomienda. Aparte también hay que recordar que después el fichero `hosts` también se va a enviar a los clientes. En el caso del cliente también es importante configurarlo, para que sea capaz de comunicarse con el servidor antes de que le mande el fichero.

El fichero nos debería de quedar de forma silimilar a lo que tenemos acontinuación, suponiendo que tenemos nuestro master en la ip 192.168.100.16 y que todas maquinas están conectadas a la red 192.168.100.0/24.

Figure 8: Fichero hosts antes de modificar


```
#  
# Copyright 2009 Sun Microsystems, Inc. All rights reserved.  
# Use is subject to license terms.  
#  
# Internet host table  
#  
::1 solarisserver2.nowhere.org localhost  
127.0.0.1 solarisserver2.nowhere.org localhost loghost  
192.168.100.16 solarisserver2.nowhere.org solarisserver2  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

"/etc/inet/hosts" 9L, 280B 9,56 A11

Figure 9: Fichero hosts después de modificar

9.3 Configuración del Server

Antes de empezar la configuración se recomienda haber realizado las configuraciones previas, a continuación la siguiente explicación se considera que se tiene todo bien configurado

9.3.1 Preparación previa de los ficheros

Antes de empezar con la configuración como explicábamos arriba el proceso NIS utiliza unos mapas que genera apartir de los ficheros que se encuentran en el sistema. Por una cuestión de seguridad y comodidad es recomendable preparar los ficheros de configuración de los cuales queramos generar los mapas para así poder evitar problemas.

1. Comprobar todos los ficheros que hay disponibles en `/etc`, están configurados y contienen la información que queremos
 - `audit_user`
 - `auth_attr`
 - `auto.home` o `auto_home`
 - `auto.master` o `auto_master`
 - `bootparams`
 - `ethers`
 - `exec_attr`
 - `group`
 - `hosts`
 - `ipnodes`

- netgroup
- netmasks
- networks
- passwd
- protocols
- rpc
- service
- shadow
- user_attr

2. Copia los ficheros en un directorio de tu elección distinto de `/etc`

Es recomendable poner los ficheros de configuración a un fichero distinto de `/etc`, por una cuestión de seguridad. La carpeta puede estar en cualquier carpeta del sistema de ficheros, pero es recomendable que sea una que tenemos controlado. Por ejemplo, una buena opción sería guardar en `/var/yp` **También es una buena practica eliminar tanto del fichero group, shadow y passwd la entrada que corresponde a root, ya que si no estaríamos compartiendo con todos los usuarios que se conecten a nuestro servidor, información relacionada con el root del sistema. Aunque destacar que el más importante es el shadow**

OJO el fichero aliases que se encuentra en `/etc/mail/aliases`, para obtener más información de la razón por la que no se puede mover el fichero de la carpeta `/etc/mail`, consultar `aliases(4)`

3. Limpiar todos los comentarios o otras líneas extrañas que puedan contener los ficheros

Al ejecutar generar los mapas normalmente ya borra los comentarios pero por seguridad y comodidad es una buena practica eliminarlos manualmente o utilizando algún tipo de script.

4. Asegurarse que la información que hay guarda en los ficheros está correctamente formateada

9.3.2 Preperación y configuración del Makefile

Antes de continuar con la configuración va a ser necesario instalar unos paquetes extra. Esto será necesario si no tenemos el Makefile instalados, en caso de tenerlo alomejor es una buena opción instalarlo ya que va a hacer falta instalarlo más adelante para poder configurar el server.

```
pkg install pkg:/service/network/nis
```

Una vez instalado el paquete deberíamos de tener el Makefile en al siguiente carpeta: `/var/yp`

Antes de empezar a modificar cosas dentro del fichero sería una buena opción guardar una copia del fichero.

Habría una serie de variables que podemos modificar para indicar donde se encuetran los ficheros para luego generar los mapas como comentamos más arriba:

- DIR

Por defecto el valor que tiene es: `/etc`. Aquí habría que poner la carpeta donde tenemos guardado los ficheros

- PWDIR

Es una variable que sirve unica y exclusivamente para indicar la ruta a `shadow` y para `passwd`

- RBACDIR

Es una variable que al igual que la anterior, sirve para indicar la posición de una ficheros en concretos, es este caso son:

```
audit_user,auth_attr,exec_attr,prof_attr
```

9.3.3 Lanzar el comando de comando de configuración: ypinit

Para crear los mapas y alguna configuración extra más habría que lanzar el comando `ypinit -m`. Esto lo que va a hacer es habilitar los servicios necesarios y aparte crear los distintos mapas apartir de los ficheros que tenemos configurados en el Makefile. Aparte nos va realizar una serie de preguntar que tendremos responder durante el proceso.

```
root@solarisserver2:~# ypinit -m

In order for NIS to operate successfully, we have to construct a list of the
NIS servers. Please continue to add the names for YP servers in order of
preference, one per line. When you are done with the list, type a <control D>
or a return on a line by itself.
    next host to add: solarisserver2.nowhere.org
    next host to add: solarisserver2
    next host to add: █
```

Figure 10: Primer paso en la configuración del script ypinit

```
root@solarisserver2:~# ypinit -m

In order for NIS to operate successfully, we have to construct a list of the
NIS servers. Please continue to add the names for YP servers in order of
preference, one per line. When you are done with the list, type a <control D>
or a return on a line by itself.
    next host to add: solarisserver2.nowhere.org
    next host to add: solarisserver2
    next host to add:
The current list of yp servers looks like this:

solarisserver2.nowhere.org
solarisserver2

Is this correct? [y/n: y] █
```

Figure 11: Segundo paso en la configuración del script ypinit

```
root@solarisserver2:~# ypinit -m

In order for NIS to operate successfully, we have to construct a list of the
NIS servers. Please continue to add the names for YP servers in order of
preference, one per line. When you are done with the list, type a <control D>
or a return on a line by itself.
    next host to add: solarisserver2.nowhere.org
    next host to add: solarisserver2
    next host to add:
The current list of yp servers looks like this:

solarisserver2.nowhere.org
solarisserver2

Is this correct? [y/n: y]

Installing the YP database will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n] █
```

Figure 12: Tercer paso en la configuración del script ypinit

```
There will be no further questions. The remainder of the procedure should take
5 to 10 minutes.
Building /var/yp/sales.nowhere.org/ypservers...
Running /var/yp /Makefile...
updated passwd
updated group
updated hosts
updated networks
updated netgroup
/usr/sbin/makedbm /var/yp/configfiles/netmasks /var/yp/'domainname'/netmasks.byad
ddr;
updated netmasks
updated auto.master
updated auto.home

solarisserver2.nowhere.org has been set up as a yp master server without any err
ors.

If there are running slave yp servers, run yppush now for any data bases
which have been changed. If there are no running slaves, run ypinit on
those hosts which are to be slave servers.
root@solarisserver2:/var/yp# █
```

Figure 13: Cuarto paso y ultimo es la salida final una vez que termina ypinit

Al tenerminar nos va generar una carpeta en `/var/yp/domainname`, donde se van a guardar todos los mapas una vez que acabe el pequeño proceso de configuración que se lanza.

9.3.4 (OPCIONAL) Configurar el fichero securenets

Aunque al configurar el servidor no recibamos ningún error y después de configurar el cliente todo funcione correctamente si desde la maquina que hace como servidor lanzamos el comando `dmesg` vamos a ver que nos devuelve el siguiente error:

```
Apr  7 18:15:57 solarisserver2.nowhere.org ypxfrd[1402]: [ID 783678 daemon.warni
ng] /usr/lib/netsvc/yp/ypxfrd: no /var/yp/securenets file
root@solarisserver2:/var/yp#
```

Figure 14: Error que se lanza porque se detecta que el fichero securenets no está creado

Esto es debido a que lo que se configura en este fichero es a que redes o a que maquinas permitimos conectarse a nuestro servidor. Por defecto viene desabilitado, pero es una buena practica el configurarlo, porque sirve como una primera capa de protección para nuestro servidor. Es cierto que si no se configura el fichero no sé va a comprobar pero si lanza el error. En el momento que lo creemos se empezar a utilizar.

Es por ello que vamos a ver como crearlo y como añadir las entradas que queramos.

Para crear el fichero habria varias opciones o directamente se crea al abrirlo con un editor de texto `vi`, `nano`, `vim`, ... o por lo contrario haciendo un *touch* y luego abriendolo con un editor de texto.

En este caso la sintaxis del fichero es la siguiente:

maska de red ip

En caso de que solo querramos permitir el acceso a un host, tenemos dos opciones:

1. 255.255.255.255 ip
2. host ip

En este caso poner host es equivalente a poner 255.255.255.255. Por cada host o cada red que queramos añadir tenemos que poner una entrada en este fichero. Obviamente no hay que olvidarse de añadir también la interfaz de localhost para evitar problemas. Un ejemplo podría ser, suponiendo que ambas maquinas estén en al siguiente red: 192.168.100.0/24

[illegible]

Figure 15: Fichero de configuración securenets

Después de crearlo sería una buena opción reiniciar los servicios como siempre en el servidor para asegurarse que se aplican los cambios.

```
svcadm refresh network/nis/domain
svcadm refresh network/nis/server
```

9.4 Configuración del Slave

Cuando se configura un servidor NIS es posible, aunque no necesario, servidores slave o esclavos que almacenen también los mapas que se encuentran en el servidor y que puedan servirlos a los clientes. Servirían para asegurar control contra fallos y para mejorar la eficiencia, ya que si tenemos muchos clientes alomejor que solo hubiera un servidor respondiendo podría saturar y provocar que se callera el servicio o una latencia incomoda.

Para ello vamos a ver como configurar un slave server.

9.4.1 Configurar primero el slave como cliente

Antes de poder configurar la maquina como slave no lo podemos hacer si no tenemos los mapas para ellos vamos a convertir la maquina primero en un cliente para obtener todos los mapas, para ello vamos a realizar igual que arriba, el comando `ypinit -c`

Es importante que cuando nos pida el nombre de los distintos servidores pongamos de primero el nombre del slave, después en segundo lugar el nombre del master en el dominio y después por último el resto de slave en caso de que los haya

9.4.2 Descargar paquete de configuración

Para poder ser slave server al igual que en el master server es necesario descargarse el siguiente paquete:

```
pkg install pkg://solaris/service/network/nis
```

Es necesario instalarlo aunque es cierto que es poco probable que usemos el Makefile o otras utilidades, aunque puede darse la necesidad. Lo principal del paquete es descargar el servicio que permite responder peticiones este sería `network/nis/server` que sin paquete no lo tenemos en el sistema

9.4.3 Iniciar la maquina como un slave server

Para ello vamos a utilizar el siguiente comando

```
ypinit -s
```

Una vez configurado ya podríamos ponerlo dentro de la lista de servidores disponibles para los clientes. El problema ahora es que cada vez que toquemos los mapas en el master vamos a tener que mandarselo a todos los slave

Igual que en el server al tener que usar el servicio `network/nis/server` que por debajo utiliza `rpcbind`, vamos a tener el mismo problema que nos encontramos antes. Cuando el cliente se intente comunicar con nuestro slave server este no le va a responder porque no va a procesar las peticiones para ello igual que hicimos en el server es necesario habilitar el procesamiento de las peticiones

```
svccfg -s svc:/network/rpc/bind setprop config/local_only=false
svcadm refresh svc:/network/rpc/bind
svcadm restart svc:/network/rpc/bind
```

También al igual que en el caso del master a un slave también se le puede configurar el fichero `securenets` para así poder controlar un poco quien se puede conectar con nuestro servidor

9.5 Configuración del Cliente

En este caso después de realizar los pasos previos que comentaba anteriormente sería un proceso super sencillo una vez que tenemos el fichero `nsswitch` modificado para que realice las búsquedas en el servidor NIS, de que tengamos el dominio NIS configurado y añadido el servidor al fichero `/etc/inet/hosts`. Solo nos quedaría lanzar el comando: `ypinit -c`.



```
root@solariscliente3:~# ypinit -c

In order for NIS to operate successfully, we have to construct a list of the
NIS servers. Please continue to add the names for YP servers in order of
preference, one per line. When you are done with the list, type a <control D>
or a return on a line by itself.
    next host to add: █
```

Figure 16: Primer paso del proceso de configuración de `ypinit` en el cliente

```
root@solariscliente3:~# ypinit -c

In order for NIS to operate successfully, we have to construct a list of the
NIS servers. Please continue to add the names for YP servers in order of
preference, one per line. When you are done with the list, type a <control D>
or a return on a line by itself.
  next host to add: solarisserver2.nowhere.org
  next host to add: solarisserver2
  next host to add: █
```

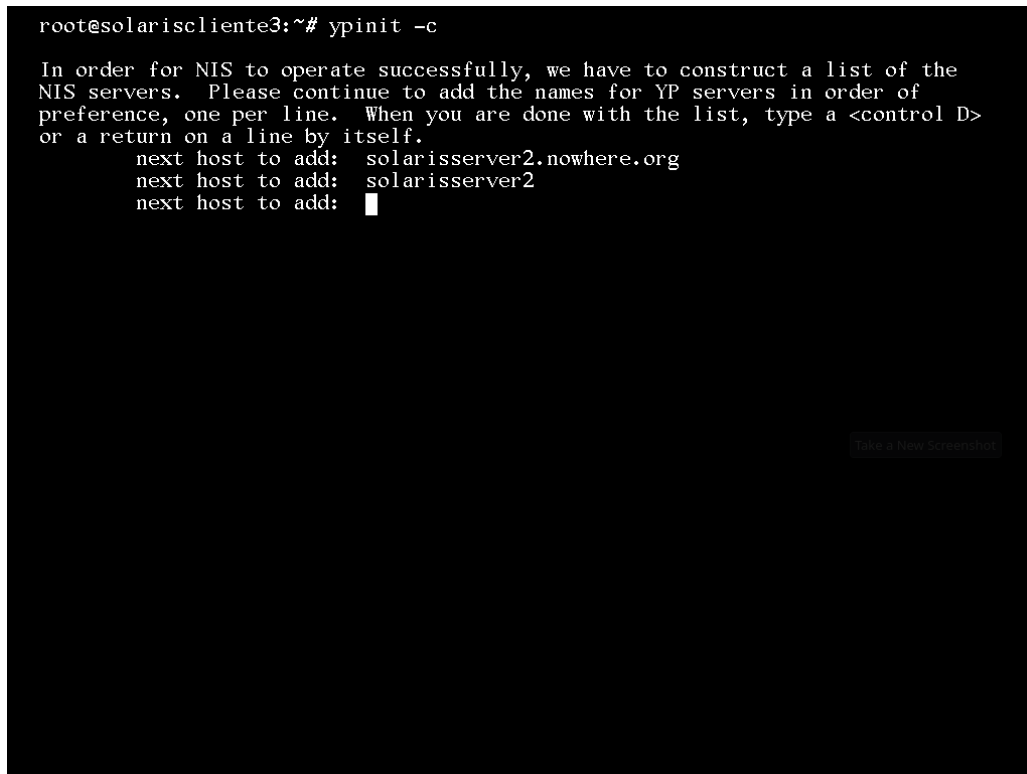


Figure 17: Segundo paso del proceso de configuración de ypinit en el cliente

Una vez terminado el proceso lo único, aunque no lo parezca lo que hizo aparte de arrancar los servicios para que todo funcione correctamente. También crea la siguiente carpeta `/var/yp/domainname/` que dentro tiene una fichero con el nombre `ypservers` que contiene un listado de los distintos servidores a los que se va a intentar conectar el cliente. Esto van a ser los mismo que se indican cuando se lanza el script cuando ponemos `ypinit -c` que quedaría sería probar que tuvieramos conexión esto lo podríamos probar realizando el comando `ypwhich`.

9.5.1 Problemas conexion cliente con el servidor

```
root@solariscliente3:~# ypwhich
Domain sales.nowhere.org not bound on solariscliente3.nowhere.org.
root@solariscliente3:~# Apr  7 17:59:17 solariscliente3.nowhere.org ypbindf11801
: NIS server not responding for domain "sales.nowhere.org"; still trying
root@solariscliente3:~# █
```




Figure 18: Problema de conexión del cliente con el servidor después de configurarlo

A veces puede pasar que después de configurar el cliente intentemos hacer alguno de los comandos para probar la conexión que comentaba anteriormente y que nos de un error como el de la foto. Para solucionar este, puede deberse a muchas cosas. Pero uno de los problemas puede ser debido a que el servicio que se encarga de procesar las peticiones en el cliente que es ypserv, por debajo utiliza rpcbind. Si no sé modifica por defecto viene configurado para solo aceptar peticiones de si mismo pero no de otras que le llegan por otras interfaces. Esto puede producir que cuando cliente intenta comunicarse con el servidor rechace la conexión. Este se puede comprobar si en la maquina servidor ejecutamos el siguiente comando `dmesg`

```
Apr  7 17:59:08 solarisserver2.nowhere.org rpcbind: [ID 702911 daemon.warning] r
efused connect from 192.168.100.15 to getaddr()
Apr  7 17:59:16 solarisserver2.nowhere.org last message repeated 11 times
root@solarisserver2:~#
```

Figure 19: Mensaje de error en el servidor cuando el cliente se intenta conectar

Por tanto para solucionar este problema y poder comunicarnos con el servidor lo que tendríamos que hacer sería utilizando el manejador de servicios de solaris modificar esa propiedad. Por tanto habría que seguir un proceso similar a cuando modificamos el fichero `nsswitch` y cuando añadimos el nombre del dominio.

Los comandos que se muestran a continuación obviamente habría que ejecutarlos en el servidor.

```
svccfg -s svc:/network/rpc/bind setprop config/local_only=false
svcadm refresh svc:/network/rpc/bind
svcadm restart svc:/network/rpc/bind
```

Aparte también habría que reiniciar los servicios que están relacionados con el NIS para que detecten la nueva configuración. Aunque a veces puede no hacer falta es una buena practica.

```
svcadm refresh network/nis/domain
svcadm refresh network/nis/server
```

Ahora podríamos lanzar otra vez el `ypwhich` y ver lo que pasa

```
root@solariscliente3:~# ypwhich
solarisserver2.nowhere.org
root@solariscliente3:~#
```

Figure 20: Solución al problema de conexión desde el cliente

Puede ser que alomejor la primera vez que lancemos el comando `ypwhich` en el cliente después de toda la configuración nos vuelva a mostrar el mismo error, pero eso puede a pasar a veces si no dejamos el suficiente

tiempo para que se actualize, si volvemos a lanzarlo un par de veces vamos a ver como nos devuelve el mensaje que esperamos.

10 Que es NFS

NFS (Network File System) es un protocolo que permite que varios ordenadores dentro de una red accedan y compartan archivos como si estuvieran en su propio sistema. Es decir, hace posible trabajar con archivos remotos como si fueran locales.

Fue desarrollado por Sun Microsystems en 1984, y funciona de forma independiente del tipo de máquina o sistema operativo, ya que utiliza estándares como XDR (para la presentación de datos) y ONC RPC (para la comunicación entre procesos).

NFS viene incluido por defecto en muchos sistemas UNIX y distribuciones Linux.

10.1 Como funciona

El sistema NFS se basa en una arquitectura cliente-servidor:

- El servidor NFS almacena los datos de forma centralizada.
- Uno o más clientes NFS acceden remotamente a esos datos a través de la red.

Gracias a esta estructura:

- Se reduce el uso de espacio en disco en los clientes, ya que los datos no se replican localmente.
- Los usuarios pueden acceder a sus directorios personales (“home”) desde cualquier máquina de la red, ya que estos se alojan directamente en el servidor NFS.

11 Como configurar NFS en Ubuntu Server

11.1 Configuración del servidor

11.1.1 Instalación de paquete necesarios

Para poder configurar el server sería necesario descargar una serie de paquetes, para ello tendríamos que lanzar el siguiente comando desde terminal:

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

Una vez instalado, habría que iniciar el servidor de NFS lanzado desde la terminal el siguiente comando:

```
sudo systemctl start nfs-kernel-server.service
```

11.1.2 Configuración de los directorios exportados

Para indicar que directorio vamos a querer configurar habría que editar el siguiente fichero `/etc/exports`. A continuación se muestra un ejemplo de una posible configuración:

```
/svr      *(ro, sync, subtree_check)

/home     *.hostname.com(rw, sync, no_subtree_check)
```

De la configuración habría que detallar algunas cosas:

- Directorio exportado: Especifica la ruta del directorio que se compartirá, en este ejemplo se comparten los siguientes: `/svr`, `/home`
- Restricciones a los clientes: El asterisco (*) o el uso de patrones (por ejemplo, `*.hostname.com`) determina qué máquinas pueden montar el recurso.
- Opciones de montaje:
 - `ro/rw`: Define si el directorio se monta como de solo lectura (`ro`) o lectura-escritura (`rw`).
 - `sync/async`: La opción `sync` garantiza que los cambios se escriban en el almacenamiento estable antes de responder a las peticiones, lo que aporta mayor seguridad, mientras que `async` mejora el rendimiento a riesgo de pérdida o corrupción de datos en caso de fallo.
 - `subtree_check / no_subtree_check`: Estas opciones controlan la verificación de que, al intentar montar subdirectorios, se tiene permiso para acceder a ellos. Aunque `subtree_check` añade una capa de seguridad, puede impactar el rendimiento en usos intensivos (por ejemplo, directorios personales con renombrado frecuente de archivos).
 - `no_root_squash`: Permite que un usuario `root` de la máquina cliente pueda modificar archivos que son propiedad del usuario `root` en el servidor. Esta opción mejora la conveniencia, pero puede generar problemas de seguridad en entornos con múltiples usuarios.

Una vez que tengamos configurado el fichero como consideremos lo que tendremos que hacer sería aplicar la configuración ejecutando el siguiente comando:

```
sudo exportfs -a
```

11.2 Configuración del cliente

11.2.1 Instalación de paquetes necesarios

Para habilitar el soporte de NFS habría que descargar el siguiente paquete: `nfs-common`. Para ello vamos a ejecutar el siguiente comando:

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

11.2.2 Montaje del directorio manualmente

Destacar que el directorio que queremos utilizar para realizar el montaje debe de estar creado, en caso de que no lo este habría que ejecutar desde la terminal:

```
sudo mkdir <nombr carpeta>
```

Una vez que tenemos decidida la carpeta lo que tendríamos que hacer sería utilizar el comando `mount`. De forma que en la terminal tendremos que introducir el siguiente comando:

```
sudo mount example.hostname.com:/srv /opt/example #en vez del hostname podríamos poner la ip
```

11.2.3 Montaje automatico mediante /etc/fstab

Para poder asegurarnos que se monte automáticamente tendríamos que modificar el fichero `/etc/fstab`. Un ejemplo de configuración podría ser algo como el siguiente:

```
#En vez del hostname también se podría poner directamente la ip
example.hostname.com:/srv /opt/example nfs defaults 0 0
```

Acerca de la configuración anterior cabe resaltar:

- Servidor y ruta exportada: `example.hostname.com:/srv`
- Punto de montaje local: `/opt/example`
- Tipo de sistema de archivos: `nfs`

Se podrían añadir otras opciones de montaje pero para el caso son irrelevantes y se dejan por defecto.

12 Como configurar NFS en FreeBSD

12.1 Configuración del servidor

Para configurar el servidor, debemos añadir las siguientes tres líneas al fichero `/etc/rc.conf`:

```
mountd_enable="YES"
mountd_flags="-r"
nfs_server_enable="YES"
```

Figure 21: Salida del fichero `rc.conf` del servidor de NFS en FreeBSD

De esta forma, activamos el servicio `mountd`, responsable de gestionar las peticiones de montaje. Colabora con `nfsd` y se encarga de verificar permisos y exportaciones en el momento en que un cliente intenta montar un sistema de archivos NFS. Le añadimos el flag `-r` para que se recargue la configuración al hacer cambios en el archivo `/etc/exports`.

Con la tercera línea, activamos el servicio `nfsd`, que se encarga de atender las peticiones de los clientes NFS, gestionando el acceso remoto a los archivos compartidos.

```
$ cat /etc/exports
/home -alldirs -maproot=root -network 192.168.100.0 -mask 255.255.255.0
```

Figure 22: Salida del fichero `exports`

Este sería un ejemplo del fichero `/etc/exports`, en el que estamos exportando el directorio `/home` a la red `192.168.100.0/24`.

El flag `-alldirs` otorga acceso a todos los subdirectorios dentro de `/home`, y `-maproot=root` mapea al usuario `root` del cliente como `root` en el servidor NFS, dándole así acceso completo al recurso compartido.

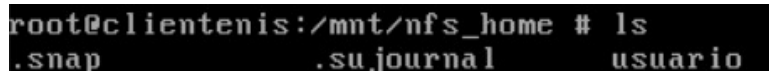
12.2 Configuración del cliente

Para el cliente, debemos tener iniciado el servicio rpcbind, que ya debería estar activo tras configurar el cliente NIS (si estamos usando la misma máquina).

En nuestro caso, creamos un directorio `/mnt/nfs_home`. Entonces ejecutamos el siguiente comando:

```
mount -t nfs 192.168.100.4:/home /mnt/nfs_home
```

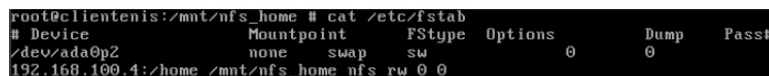
Tras esto, si listamos los contenidos de este directorio, deberíamos tener acceso a los ficheros y directorios del directorio `/home` del servidor.



```
root@clientenis:/mnt/nfs_home # ls
.snap          .su journal    usuario
```

Figure 23: Salida del fichero exports

Una vez vemos que está todo correcto, podemos añadir una entrada al fichero `/etc/fstab`. De esta forma, se montará el directorio automáticamente al iniciar la máquina.



```
root@clientenis:/mnt/nfs_home # cat /etc/fstab
# Device            Mountpoint          FStype  Options    Dump    Pass#
/dev/ada0p2         none               swap    sw         0       0
192.168.100.4:/home /mnt/nfs_home      nfs     rw         0       0
```

Figure 24: Salida del fichero exports

13 Como configurar NFS en Solaris

13.1 Configuración del server

Simplemente habría que hacer un sencillo paso que sería deducir la carpeta que vamos a exportar y una vez decidido tendríamos que ejecutar el siguiente comando:

```
share -F nfs -o rw <path/nombrecarpeta>
```

En el caso de `[path/nombrecarpeta]` hace referencia a la carpeta que queremos compartir, ahí tendríamos que poner la carpeta que nosotros queremos compartir.

Destacar también que en este caso se está compartiendo la carpeta con permisos de lectura escritura, pero también se podría configurar la carpeta de solo lectura. Incluso también se podría hacer unas listas de control de acceso. Pero para más información se recomienda consultar `man share`

También para saber si estamos compartiendo una carpeta o para saber si una vez ejecutado el comando todo está funcionando bien lo que se podría hacer sería:

```
share -F nfs
```

13.2 Configuración del cliente

Para configurar el cliente en NFS es un proceso muy sencillo, en este caso habría dos opciones, una que es la más interesante para hacer que siempre se nos configure al arrancar o por el contrario se puede hacer temporal.

1. Para hacer que la configuración sea temporal simplemente habría que hacer un mount de la siguiente manera:

```
mount -F nfs <ip>:/<path + nombrecarpeta> <carpetamontaje>
```

En este caso la *ip* sería la del servidor que está ofreciendo el servicio NFS y después *path + nombrecarpeta* es la carpeta que se está compartiendo. Finalmente el cliente tendría que decidir la carpeta donde se quiere montar, esto es lo que se corresponde con *carpetamontaje*

2. Para hacer que la configuración no sea temporal y para que siempre que arrancamos la maquina nos configure todo de manera correcta:

Habría que realizar dos pasos muy sencillos:

- (a) Hay que añadir una entrada al fichero `/etc/vfstab`

El siguiente fichero se utiliza para determinar que sistemas de ficheros quieres que se monte durante el arranque. En este caso sigue un formato de tabla separado por espacios, de forma que si quisieramos añadir una entrada quedaría de la siguiente manera:

Es importante que los parametros estén separados por espacios para evitar errores.

Para obtener más información se recomienda mirar `man vfstab`

- (b) Después de añadir la entrada es necesario reiniciar el servicio para asegurarnos que se guarda el cambio:

```
svcadm enable -r nfs/client
```

14 Inteconexión NIS y NFS

Es muy habitual que en una red en la que se configura un servicio NIS también se configure un servicio NFS, para así poder conseguir centralizar y controlar la administración de la red. En consecuencia de la gran sinergia que tienen ambos sistemas. Ya que desde NIS podemos controlar los puntos de montaje, los usuarios y los grupos que puede acceder, a las carpetas que exportamos desde nuestro servicio de NFS.

Normalmente dependiendo de la magnitud de la empresa se pueden tener en cuenta más ficheros de configuración o menos. Un ejemplo donde se podría ver el potencial de esta configuración, sería por ejemplo, en una empresa o en una universidad.

Supongamos que tenemos una empresa donde tenemos dos areas de trabajo finanzas y contabilidad. Pues lo que podríamos hacer sería, configurar dos dominios NIS, una para la parte finanzas y otro para el departamento de contabilidad. Y después cada vez que llega un empleado nuevo le podemos crear un usuario dentro de su dominio, de forma que cuando llegue a su maquina sea capaz de loggearse sin problema. Aparte como tenemos configurado un servidor NFS vamos a poder crearle una carpeta al nuevo usuario para que así el pueda guardar sus archivos en su propia carpeta. De esta forma cualquier usuario de el area de finanzas podría utilizar cualquier ordenador que estuviera dentro del dominio NIS que le corresponde a esta departamento, y podría entrar con su usuario y acceder a sus archivos sin ningún tipo de problema. obviamente esto se podría mejorar y dependiendo del sistema en el que estén corriendo los clientes, podríamos, por ejemplo, si fuera Solaris, se podrían crear un mapa del fichero `/etc/vfstab` para controlar los puntos de montaje en las distintas maquinas, o cualquier otro tipo de necesidad que tuvieramos

15 Comparación entre NIS,NIS+ y LDAP

Aunque NIS es una muy buena opción y muy sencillo de configurar es cierto que tiene bastantes problemas de seguridad que si es cierto que hay muchas opciones y tecnicas que se podrían hacer para securizarlo o por lo

menos asegurar que todo este más protegido. La realidad es que hoy en día existen alternativas más modernas como puede LDAP, o una opción de NIS+ que trae una capa de securización mayor.

Es por ello que a continuación se ofrece una pequeña comparativa entre ambos.

15.1 Seguridad

- **NIS:** No implementa seguridad criptográfica. No hay cifrado de datos ni autenticación de origen; cualquier cliente dentro del dominio puede consultar los datos. Los hashes de contraseña y demás información sensible están expuestos en los mapas NIS, accesibles a cualquiera con acceso a la red interna. En la práctica, NIS confía en la seguridad física o perimetral de la red (lo cual se considera inadecuado hoy en día).
- **NIS+:** Introdujo mejoras significativas en seguridad. Utiliza DES para autenticar y cifrar las comunicaciones mediante el mecanismo de Secure RPC (`AUTH_DES`). NIS+ maneja credenciales para clientes y servidores, requiriendo que ambos se autenticuen mutuamente antes de intercambiar datos. Además, permite definir permisos de acceso granulares a los objetos del directorio (tablas NIS+) basados en identidades y grupos, evitando que usuarios no autorizados consulten cierta información. Estas medidas cierran muchas brechas de NIS, aunque a costa de mayor complejidad administrativa. (*Nota: Pese a la mejora, desde 56 bits hoy se considera criptográficamente débil, pero en el contexto original de NIS+ era un gran avance sobre la ausencia total de cifrado de NIS*).
- **LDAP:** Fue diseñado con la seguridad en mente desde el inicio. LDAP puede operar sobre SSL/TLS, proporcionando cifrado fuerte de todo el tráfico (ej.: LDAP con `StartTLS` o `LDAPS`). También soporta múltiples esquemas de autenticación (simple con usuario/contraseña, SASL con mecanismos como GSS-API/Kerberos, etc.). Además, LDAP implementa control de accesos a nivel de entrada y atributo: el administrador puede definir políticas precisas sobre quién puede leer o modificar cada dato. En resumen, LDAP ofrece autenticación robusta y cifrado (según configuración), superando por mucho las capacidades de seguridad de NIS y también por encima de NIS+ en flexibilidad (pudiendo usar algoritmos modernos y autenticación integrada con Kerberos, certificados, etc.).

15.2 Escalabilidad y arquitectura

- **NIS:** Está pensado para redes locales (LAN) relativamente pequeñas o medianas. La arquitectura de NIS es de dominio plano: un dominio NIS abarca una serie de máquinas, pero NIS no establece jerarquías entre dominios (no hay dominios padres/hijos). Esto limita su escalabilidad; por ejemplo, organizaciones grandes tendrían que gestionar múltiples dominios NIS separados sin una forma sencilla de conectarlos. Además, NIS utiliza difusión (*broadcasts*) para descubrir servidores en la red local (a menos que se configuren explícitamente en cada cliente), lo que no escala bien en subredes grandes o segmentadas por ruteo.
- **NIS+:** Diseñado para ser más escalable y flexible que NIS. Soporta un espacio de nombres jerárquico, similar al de DNS, con un dominio raíz y subdominios organizados posiblemente por regiones o departamentos. Esto permite delegar partes del directorio a diferentes servidores y manejar entornos corporativos más grandes de forma estructurada. Cada subdominio NIS+ puede tener su propio maestro y réplica, mejorando la distribución de carga. No obstante, en la práctica NIS+ no alcanzó despliegues masivos: su complejidad hizo que muchas organizaciones pequeñas no lo adoptaran, y las muy grandes tendieron a migrar a LDAP en cuanto estuvo disponible, por su aún mayor capacidad. En resumen, NIS+ es más escalable que NIS (soporta múltiples dominios jerárquicos), pero finalmente está limitado al ámbito empresarial y ha sido superado por LDAP y otros servicios globales.
- **LDAP:** Es altamente escalable a nivel global. El directorio LDAP sigue un esquema jerárquico (árbol DIT), generalmente alineado con el DNS de la organización y puede distribuirse en múltiples servidores mediante replicación multi-maestro si es necesario. LDAP fue diseñado para soportar millones de entradas distribuidas mundialmente, siendo el estándar en entornos corporativos extensos e incluso para servicios de directorio de proveedores (como directorios telefónicos, etc.). Además, LDAP opera sobre TCP/IP estándar, lo que lo hace amigable con las infraestructuras de red actuales y facilita su funcionamiento a través de Internet o WAN (a diferencia de NIS, que esencialmente no funciona fuera de la LAN sin tunelado). En cuanto a rendimiento, LDAP puede indexar atributos y manejar búsquedas eficientes incluso en directorios muy grandes. Por ello, en escalabilidad, LDAP supera ampliamente a NIS y NIS+.

15.3 Facilidad de administración

- **NIS:** Es relativamente simple de configurar comparado con las alternativas. La sintaxis y concepto son básicos (mapas generados desde archivos planos) y no requiere diseño de esquemas ni planificación de árboles. Herramientas como `ypwhich`, `ypcat` facilitan la verificación. Por ello, en entornos pequeños, NIS podía ponerse en marcha rápidamente y con poca sobrecarga administrativa.
- **NIS+:** Sacrifica simplicidad a cambio de seguridad y funcionalidad. Más difícil de configurar y administrar que NIS, ya que requiere establecer un dominio de directorio seguro, generar credenciales para servidores y clientes (usando `nisaddcred` en Solaris), y entender su modelo de permisos. Administrar NIS+ implica manejar tablas en lugar de simples mapas, y comandos más complejos (`nistbladm`, `nisadm`, ...). La percepción general fue que NIS+ era "seguro pero difícil", en contraste con NIS "inseguro pero fácil".
- **LDAP:** Requiere una curva de aprendizaje mayor y una planificación cuidadosa, especialmente en entornos nuevos. Configurar un servidor LDAP (como OpenLDAP) implica definir un esquema (atributos y objetos) adecuado para su organización, pensar la jerarquía del DIT, y configurar políticas de acceso. También hay que integrar servicios (por ejemplo, NSS/PAM en Linux, o aplicaciones) para que usen LDAP. Todo esto puede resultar complejo y exige conocimientos especializados. Sin embargo, con herramientas modernas y amplia documentación, LDAP se ha vuelto más accesible con el tiempo. Hoy existen soluciones integradas (`389-DS`, `FreeIPA/IdM`, `OpenLDAP` con *frontends*, `Apache DS`, etc.) que simplifican la administración proporcionando interfaces o automatizando muchas tareas (como agregar usuarios, gestionar políticas).

15.4 Soporte actual y estado de la tecnología

- **NIS:** Es considerado legado en los sistemas actuales. Las distribuciones Linux aún proveen paquetes de NIS (por compatibilidad con instalaciones antiguas), pero varios fabricantes han anunciado su desuso. Por ejemplo, Red Hat marcó NIS como *deprecated* en RHEL 8.3 y lo eliminó completamente en RHEL 9. Esto se debe a sus deficiencias de seguridad y a que existen alternativas superiores. En entornos profesionales, es raro desplegar un nuevo servicio NIS; solamente se mantiene donde ya existía por compatibilidad hacia atrás.
- **NIS+:** Tuvo soporte oficial en Solaris hasta la versión 10, pero fue discontinuado en Solaris 11 en 2012. Oracle recomienda migrar cualquier entorno NIS+ restante hacia LDAP. En Linux, nunca tuvo un soporte completo de primera clase (existieron implementaciones parciales como librerías NSS para NIS+, pero no fue ampliamente adoptado). Al día de hoy, NIS+ prácticamente no se utiliza; es visto como un producto obsoleto de los 90. Sus aportes en seguridad fueron importantes en su momento, pero no lograron sostener su vigencia.
- **LDAP:** Es la tecnología vigente y recomendada para servicios de directorio. Todos los sistemas operativos modernos tienen soporte LDAP (ya sea vía servicios nativos o integraciones). LDAP actúa como base de soluciones de identidad más amplias, por ejemplo *Active Directory* de Microsoft está basado en LDAP (con extensiones) y es omnipresente en entornos empresariales. Hoy día, LDAP cuenta con amplio soporte y además, continúa evolucionando, soportando cifrados modernos, mejoras de rendimiento, etc. Dado que LDAP es un estándar abierto (RFC 4510+), su soporte está garantizado a futuro y su interoperabilidad es muy alta, a diferencia de NIS/NIS+ que eran más específicos de ciertos entornos.

En resumen LDAP en la mayoría de aspectos es mucho mejor que NIS+ y que NIS, aunque lo que sí que tiene de bueno NIS es que es bastante simple. Y podría ser útil alomejor en un entorno controlado o incluso en uno doméstico, donde es complicado tener problemas de seguridad. Aunque a nivel de empresas es mucho más recomendable el uso de LDAP, debido a toda la tecnología que implementa.

15.5 Migración de NIS a LDAP

Destacar que aunque alomejor NIS sí que puede ser una tecnología antigua existe la posibilidad de si tenemos una configuración hecha, el pasar a utilizar LDAP. Todos los sistemas operativos explicados en este documento

tienen esta posibilidad aunque el proceso es bastante complejo y se escapa del tema. Así que recomiendo al lector que consulte otras fuentes. Aunque sí que es interesante que sepa que existe esta posibilidad.