Laboratorio Sesión 09: Servicios centralizados

En esta sesión vamos a ver otro grupo de servidores de aplicaciones, esta vez centrados en la gestión centralizada de usuarios. Veremos dos servidores remotos pensados para redes de área local: el servidor de ficheros `Samba` que sirve para integrar servidores y clientes Windows y Linux y `NFS`. Estos servidores junto con `NIS` se utilizan en redes de área local para la gestión centralizada de usuarios. Dado que todos los servidores que se utilizan en esta práctica sobrecargan bastante el sistema, es bueno que al pasar de un apartado a otro paréis los demonios que habéis ido activando.

9.1. El servidor de ficheros `Samba`

El sistema `Samba` es el sistema de compartición de recursos (áreas de disco e impresoras) a través de redes de área local que utilizan los sistemas operativos de tipo Windows. Este sistema era propio de dichos sistemas y está bastante limitado, sin embargo, es ampliamente utilizado debido a la gran difusión y sencillez de uso de los sistemas Windows (es muy fácil compartir una carpeta desde Windows). A cambio de esto es un sistema altamente inseguro y por tanto no es NADA recomendable usarlo en una red abierta hacia Internet.

Debido a su gran popularidad, el sistema `Samba` ha sido totalmente adaptado a los sistemas Linux, de forma que un sistema Linux puede ser tanto cliente como servidor `Samba` (y por tanto puede pedir o servir recursos tanto a sistemas Windows como a otros sistemas Linux que también utilicen `Samba`).

En el sistema que tenéis en el laboratorio no disponéis de servidor `Samba`, pero podéis instalarlo con la herramienta gráfica. Hacedlo (acordaros de actualizar el repositorio antes).

El sistema `Samba` consta de dos demonios: `smbd` y `nmbd`. El primero proporciona los servicios de ficheros e impresoras, mientras que el segundo es el gestor de nombres del sistema usado por Windows. De todas formas, para iniciar ambos servicios disponemos del script `samba` que se encuentra, como todos, en `/etc/init.d`. Sin embargo no iniciéis el sistema ahora, leeros toda esta sección antes de hacer nada ya que antes de iniciar el script hemos modificar sus parámetros de configuración mediante el fichero `smb.conf`.

Pregunta 9.1: *¿Cómo se llama el paquete que habéis instalado? ¿Donde se ubica el fichero* `smb.conf`*?*

Esta práctica la haréis con los compañeros del ordenador de al lado (es decir, en grupos de 4), de forma que unos tengáis el servidor y otros el cliente. Primero hablad con ellos y decidid quien es el servidor y quien el cliente (a continuación podréis hacerlo al revés).

9.1.1. ¿Qué han de hacer los que tengan el servidor?

En primer lugar renombrad el fichero `smb.conf` a `smb.conf.old` y cread un nuevo fichero de configuración desde cero con el siguiente contenido:

```
[global]
workgroup = MIDEARTH
netbios name = HOBBIT
security = share

[data]
comment = Data
path = /export
read only = Yes
guest ok = Yes
```

Pero OJO: debéis usar como ‘`workgroup` OTRO nombre (de 8 letras) que no coincida con nadie más del aula (no debería ser difícil) y lo mismo para el `netbios name` (si hay más de un servidor con el mismo nombre da conflictos).

La sección `data`, ya puede ser la misma para todos, en esta sección se especifica qué es lo que se va a compartir.

Pregunta 9.2: *¿Qué directorio compartiréis?*

Una vez hayáis cambiado el `smb.conf` deberéis crear el directorio a compartir y asignarlo al usuario `alumne`, grupo `alumne` (no es bueno compartir nada de `root`, de hecho tampoco lo es dejar que el `root` se conecte). Dentro de él podéis crear algún fichero de prueba. Y entonces ya podéis iniciar el servidor `samba` igual que iniciabais el servidor `ftp` la semana pasada.

Pregunta 9.3: *¿Con qué orden se inicia el servidor* `samba`*?*

Pregunta 9.4: *¿Cómo se cambia el dueño del directorio para que pertenezca a* `alumne`*?*

9.1.2. ¿Qué han de hacer los clientes?

En primer lugar los clientes han de instalar el paquete con el servidor `samba` y parar el servicio (el instalador lo arranca automáticamente). A continuación también han de modificar su fichero `smb.conf` (renombrad el viejo para no perderlo) y dejar tan solo la definición del grupo de trabajo:

```
[global]
workgroup = MIDEARTH
```

Acordaros que ha de ser el mismo nombre que en el servidor y que NO ha de coincidir con nadie más del aula (así que NO uséis el del ejemplo).

Una vez hecho esto, en teoría, los clientes ya se pueden conectar simplemente escribiendo:

```
smbclient //<nombrePCservidor>/data
```

Sin embargo, fijaros que en el laboratorio todos los PCs tienen el mismo nombre así que en este caso conectaros usando:

```
smbclient //<IPdelPCservidor>/data
```

(Si os pide password simplemente pulsad ENTER). `Smbclient` es un cliente que funciona igual que un cliente `ftp`.

Pregunta 9.5: *¿Cómo podéis leer un fichero del servidor?*

Pregunta 9.6: *¿Podéis escribir un fichero en el servidor?*

Esta conexión es útil, pero sería más útil poder trabajar en el servidor como si de un directorio local se tratase. Para ello debéis instalar un nuevo paquete: `smbfs` (bueno, su versión actual). Una vez hecho esto, ¿os acordáis del viejo comando `mount`? Pues podéis probar a escribir esto:

```
mkdir /samba
mount -t cifs //<nombrePC>/data/ /samba -o guest,uid=1000
```

Fijaros que ya hemos compartido un directorio en modo lectura. ¿Cómo podemos ahora compartir en modo lectura/escritura?

En el servidor, modificad el fichero `smb.conf` para que se parezca a esto:

```
[global]
workgroup = MIDEARTH
netbios name = HOBBIT
security = SHARE

[data]
comment = Data
path = /export
force user = alumne
force group = alumne
read only = No
guest ok = Yes
```

A partir de ahora, si reiniciáis el servidor, el cliente se conectará como si fuera el usuario `alumne` y podrá hacer en el directorio `/export` exactamente lo mismo que podría hacer `alumne`.

Finalmente vamos a ver como se configuraría `smb.conf` para compartir los `home` de los usuarios:

```
[global]
workgroup = MIDEARTH
netbios name = HOBBIT

[homes]
comment = Home Directories
valid users = %S
read only = No
browseable = No

[public]
comment = Data
path = /export
force user = alumne
force group = alumne
guest ok = Yes
read only = No
```

Eso sí, si compartís los homes, conviene usar contraseñas. Podéis ponerle una contraseña a `alumne` para `Samba` mediante el comando `smbpasswd -a`.

Y con esto ya está. `Samba` también permite compartir impresoras, y aunque no podemos ver esta utilidad en el laboratorio podéis encontrar un montón de ejemplos útiles en `www.samba.org`.

Ahora, antes de continuar, y para no sobrecargar la red, acordaros de desmontar la unidad compartida y de parar el servidor `Samba`.

Pregunta 9.7: *¿Cómo se desmonta la unidad compartida y se para el servidor* `Samba`*?*

9.2. El servidor de ficheros NFS

El sistema `NFS` (Network File System) es el sistema de compartición de disco a través de red nativo de Linux. Es un sistema relativamente seguro y es el que se suele utilizar para crear sistemas con gestión centralizada de disco como, por ejemplo, el típico de un aula de ordenadores (donde uno se puede conectar en cualquier ordenador y acceder a su cuenta que está en un servidor central). Antes de utilizarlo deberéis proceder a instalarlo (los ordenadores que usáis no se instalaron para hacer de servidores). Para hacerlo instalad el paquete `nfs-kernel-server`.

En primer lugar debéis decidir que puntos de vuestro sistema de ficheros queréis compartir. Para ello debéis editar (o crear, si no existe) el fichero `/etc/exports` que debe contener, a razón de una por línea, una lista de aquellos directorios que queréis compartir, especificando para cada uno, con quien y con que opciones. Por ejemplo:

```
/               master(rw) trusty(rw,no_root_squash)
/projects       proj*.local.domain(rw)
/usr            *.local.domain(ro) @trusted(rw)
/home/joe       pc001(rw,all_squash,anonuid=150,anongid=100)
/pub            (ro,insecure,all_squash)
```

Si solo queréis hacer una prueba rápida podéis usar la línea:

```
/home             *(rw,no_root_squash)
```

que permitirá que los clientes lean vuestro directorio `/home`.

Pregunta 9.8: *¿Qué significan las opciones de la cuarta línea del ejemplo? ¿Y las de la línea que habéis usado?*

Una vez convenientemente creado el fichero `exports`, ya podéis arrancar el servidor `NFS` mediante el script `nfs-kernel-server` que se encuentra donde siempre.

Finalmente, podéis haceros clientes del servidor `nfs` que haya activado la otra pareja del grupo con el que trabajáis. Para ello simplemente debéis montar el directorio exportado del servidor en un directorio vacío de vuestro sistema de ficheros indicando a `mount` que el directorio es de tipo `nfs` (muy parecido al caso de `samba` pero ahora no hace falta especificar usuario y contraseña):

Pregunta 9.9: *¿Qué paquetes de Ubuntu contienen el comando* `mount.nfs`*? ¿Con qué ordenes se monta el sistema remoto en vuestro ordenador?*

Ahora ya podréis acceder al directorio según las reglas especificadas por el servidor.

Pregunta 9.10: *¿Qué permisos tenéis en el directorio remoto si os conectáis como* `alumne`*? ¿Y si os conectáis como* `root`*?*

Acordaros de desmontar y de apagar el servidor `nfs` antes de continuar.

9.3. El servidor de usuarios `NIS`

El sistema `NIS` es, en cierto modo, el complemento al sistema `NFS`. Así como `NFS` permite tener un servidor centralizado de disco, `NIS` permite tener un sistema centralizado de usuarios, es decir, gestionar los usuarios solo en el servidor central y que todo el resto de ordenadores de la red acepten los usuarios de dicho servidor central como propios. Los programas servidores vinculados a `NIS` son `ypserv` y `ypbind`.

`NIS`, además, depende del servicio `portmap` que se encuentra en el paquete `rpcbind`. Por tanto empezad instalando este paquete. A continuación instalad el paquete `nis`. Cuando os pida un dominio introducid el que queráis pero debe ser el mismo en los dos ordenadores que trabajéis juntos y distinto del de los demás. Tomároslo con calma porque tarda un rato (intenta conectarse a un servidor pero no encuentra ninguno).

Como en los casos anteriores, para ver la utilidad del sistema `NIS`, deberéis trabajar en parejas de ordenadores. Uno de los ordenadores será el servidor y el otro el cliente del sistema.

9.3.1. ¿Qué se debe hacer en el ordenador cliente?

En primer lugar deberéis especificar al sistema que hay más usuarios de los que ya tenéis configurados. Para ello deberéis modificar todos los ficheros de usuario añadiendo una última línea que contenga un `+` y tantos `:` como las anteriores (Por ej, en el fichero `passwd` deberéis añadir: `+::::::`).

Pregunta 9.11: *¿Cuáles son los 4 ficheros que debéis modificar?*

El siguiente paso es modificar el fichero `/etc/yp.conf` especificando la dirección ip del servidor y, después de que el servidor esté configurado arrancar el demonio `NIS`.

Pregunta 9.12: *¿Con qué orden se arranca el demonio* `NIS`*?*

9.3.2. ¿Qué se debe hacer en el servidor?

En el servidor también vamos a crear la configuración mínima para que funcione.

El primer paso es modificar el fichero `/etc/default/nis` especificando que el ordenador será servidor (hay que poner `master` en la opción correspondiente) y, de nuevo al igual que en el cliente, modificar el fichero `yp.conf` con la ip del servidor.

A continuación vamos a intentar arrancar `NIS` (los servicios `ypserv` y `ypbind`). Veréis que el sistema tardará un buen rato. Ahora, como ya está el servidor arrancado deberemos crear un usuario nuevo (por ej. con `adduser`). Veréis que aunque el proceso sigue siendo automático, de repente hace muchos más pasos. A continuación podéis arrancar `NIS` de nuevo (ahora sí que se arrancará todo).

Una vez hayáis finalizado los pasos, podéis probar a conectaros desde el cliente al usuario que solo habéis dado de alta en el servidor. Para hacer esto, debido (como siempre) a que no reiniciamos el sistema, probad a crear antes, por ejemplo, 3 usuarios en el servidor:

```
adduser pedro
adduser juan
adduser pablo
```

A continuación inicializad la base de datos:

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

y desde el cliente, reiniciad el servidor `nis` (para que se actualicen los datos) y probad a cambiar al usuario `juan`.

Pregunta 9.13: *¿Con qué comando se cambia de usuario en modo texto?*

Si lo habéis hecho bien os reconocerá el usuario `juan` aunque no lo hayáis creado en el cliente. Tened cuidado porque debido a que lo hemos hecho todo "en caliente" puede haber un desfase de 1 usuario entre el servidor y el cliente (es decir, si habéis seguido el orden del ejemplo, no podréis acceder a `pablo` en el cliente hasta que creéis otro usuario nuevo). Para actualizar la base de datos cada vez que hagáis un cambio ejecutad:

```
make -C /var/yp
```

Fijaros que combinando `NIS` y `NFS` se puede crear un sistema de red centralizado en el que cada usuario puede acceder a su cuenta desde cualquier ordenador de la red y, además, el mantenimientos de los usuarios (altas, bajas, etc) solo se realiza en el servidor, no en todos los clientes. Estos son simplemente clones todos con la misma configuración.

Pregunta 9.14: *Escribid las ordenes del cliente y del servidor que combinarían* `NIS` *y* `NFS` *para tener un servidor de usuarios centralizado de forma que un usuario pudiera acceder a su cuenta desde cualquier ordenador.*

Pregunta 9.15: *Tal y como está configurado el servidor* `NIS`*, enviará la información a cualquier sistema que la pida. ¿Cómo se puede limitar esto de forma que solo envíe la información a, por ejemplo, los ordenadores del aula en la que estáis?*

9.4. Los logs del sistema

Vamos a hablar ahora de una importante herramienta de auditoría que aún no conocéis: los logs del sistema. Los logs del sistema son mensajes, avisos, que nos permiten saber las cosas que han pasado (y que nos pueden parecer más o menos interesantes). Estos mensajes se recopilan siguiendo un cierto formato y se almacenan en diversos ficheros en `/var/log`. Comprobad el fichero `auth.log`. Este fichero debería contener las entradas autorizadas al sistema.

Pregunta 9.16: *¿Qué contiene el fichero* `auth.log`*? ¿Cómo se modifica su contenido después de ejecutar* `su alumne`*?*

Como veis el sistema de logs nos da información sobre lo que pasa en el sistema.

Pregunta 9.17: *¿Qué demonios gestionan en este momento los logs de vuestro sistema? (podéis encontrarlos sabiendo que están activos e incluyen la cadena* `log` *en su nombre).*

Evidentemente, la utilidad de los logs del sistema depende mucho de que estén bien configurados y que, por tanto, proporcionen información útil. Los logs dependen, tanto del programa que los envía, como del demonio `syslog` que los recoge. Este último se puede configurar mediante el fichero `rsyslog.d/50-default.conf` (en algunos ordenadores podéis tener la configuración vieja, en cuyo caso la misma información está en `syslog.conf`) que aunque a primera vista puede parecer muy complicado es bastante sencillo de configurar: los mensajes están ordenados por servicios y prioridades. En cada línea del fichero anterior la primera parte de dicha línea indica la procedencia del mensaje y actúa a modo de filtro: Si coincide con el mensaje entrante, este se guarda en el destino indicado en la segunda parte del mensaje (que puede ser un fichero, una consola u otro ordenador, de forma que podéis centralizar todos los mensajes de una sala en un solo servidor). Podéis consultar la página de `man` para más detalles (además, en la web hay documentación en castellano).

Pregunta 9.18: *Probad a crear un fichero nuevo, además de los ya existentes que recoja los mensajes de entrada al sistema (cuando un usuario se conecta). ¿Qué línea habéis introducido en en el fichero de configuración para hacerlo?*

Pregunta 9.19: *Otra posibilidad interesante con los logs es enviar vuestros logs a un ordenador que centralice las peticiones. ¿Con que línea de configuración enviaríais vuestros mensajes de entradas al sistema (los de la pregunta anterior) al ordenador de al lado? Contestad teóricamente y probadlo sólo si os sobra tiempo porque lleva bastante rato solucionar todos los problemas de permisos. Si lo conseguís avisad :-)*

Uno de los problemas posibles con los logs es que si generáis demasiada información el sistema se sature con los mensajes de log. Es por esto (y por los correos electrónicos y otros servicios) que en un servidor un poco importante es bueno tener el directorio `/var` montado en una partición distinta del disco, o incluso en otro disco físico... y claro, no olvidarse de ir echando un vistazo y borrando lo que no os interese. De todas formas, si necesitáis gestionar un sistema un poco grande y queréis olvidaros del tema pero mantener logs (la LOPD así lo pide, y es probable que os toque trabajar con ella sobre la cabeza) existe la utilidad `logrotate` que básicamente borra cada cierto tiempo los logs manteniendo copias de otra cantidad de tiempo e incluso comprimiendo los datos (pero claro, todo esto vosotros ya lo sabéis hacer mediante un script ;-).

Finalmente comentaros que no hay un único sistema de logs, hay varios (quizás el siguiente más conocido sea `syslog-ng`) y además, existen herramientas de auditoría que permiten analizar rápidamente los logs.

Pregunta 9.20: *Por cierto, en dos frases ¿qué es la LOPD? ¿qué regula? ¿cómo os afecta?*