Memoria del Trabajo de Fundamentos de Redes

Sergio Quijano Rey, Daniel Gonzálvez

Motivación

La motivación que hay detrás de este trabajo es la de estudiar cómo funcionan los productos de almacenamiento en la nube, a través de la creación de un servidor bastante básico de almacenamiento en red sobre nuestro propio servidor.

Para ello hemos usado una *Raspberry Pi 3* en la que hemos instalado el programa de software libre *Nextcloud*. Sobre este servidor muy básico hemos tenido que realizar ciertas labores de administración, más propias de una asignatura como *Ingeniería de Servidores*, y otras labores centradas en aspectos de redes.

Esto nos ha permitido conocer y aprender hasta cierto punto los aspectos a tener en cuenta para crear infraestructuras de este tipo, centrándonos especialmente en aspectos específicos de las redes.

Especificación del hardware usado

El hardware que hemos usado es el siguiente:

- Raspberry Pi 3 Model B:
  - CPU Quad Core 1,2GHz Broadcom BCM2837 64bit
  - 1GB RAM
  - Tarjeta SD *Samsumg Evo*, 64 GB (actúa como disco duro del ordenador)
- Cargador de móvil *MicroUSB*: dará corriente a la *Raspberry Pi 3*

Instalación básica del servidor

Instalación del sistema operativo

Lo primero que necesitamos es descargar el sistema operativo que vamos a usar. Nostros vamos a usar *Ubuntu Server* para la *Raspberry Pi 3*. Podemos descargar un fichero `.img` del siguiente link: Página de descarga

Una vez descargado el archivo tenemos que quemarlo en la tarjeta SD. Usando un adaptador `SD-USB` lo conectamos a nuestro ordenador. Localizamos la tarjeta SD usando el comando `lsblk`. Al ver que el dispostivo está ubicado en `/dev/sdb`, usamos el siguiente comando para quemar la imagen del sistema en la tarjeta SD: `sudo dd if="ubuntu-raspberry.img" of="/dev/sdb" bs=4M status=progress`. Una alternativa a usar la línea de comandos es usar un programa con interfaz gráfica como `etcher`.

Configuración inicial del sistema

Ahora introducimos la tarjeta SD en la *Raspberry Pi* y conectamos el cargador, tanto a la *Raspberry Pi* como a la corriente. También conectamos un cable `ethernet` desde el router hasta la *Raspberry Pi*, para tener conexión a internet

Gracias a que estamos usando *Ubuntu Server*, por defecto tiene el servicio `ssh` habilitado por defecto. Así que entramos al administrador del router escribiendo en la barra de nuestro navegador la dirección `192.168.1.1`. Tras esto podemos localizar su dirección local, como se muestra en la **Imagen 1**:

Imagen 1

Hacemos ssh a nuestro servidor con la dirección que acabamos de obtener: `ssh ubuntu@192.168.1.8`. `ubuntu` es el usuario por defecto que está creado, además de `root`. Este usuario está en la lista de `sudoers`, por lo que tenemos privilegios de administración a través de él. Pero para facilitar las cosas, nos registramos como `root` usando el comando `sudo su -`.

El primer paso es hacer una primera actualización del sistema con `apt update; apt upgrade`, e instalar algunos paquetes básicos para trabajar, como pueden ser `vim` o `make`

Creación del usuario de administración

Crearemos un usuario `administrator` para las labores de administración del servidor. Para ello ejecutamos `useradd -m administrator`. La opción `-m` crea el directorio `home` del usuario según lo especificado por el `skel`.

Añadimos al administrador al grupo `sudo` con el comando `usermod -aG sudo administrator`. La opción `-a` indica que estamos añadiendo al grupo, en vez de cambiar al usuario de grupo, y la opción `-G` es necesario para el anterior *flag*, indicando los grupos suplementarios a los que se va a añadir.

Para cambiar la contraseña al usuario, usamos el comando `passwd administrator`, que nos pide por el *prompt* introducir la antigua contraseña y, a continuación, la nueva contraseña.

Ahora borramos al usuario que venía por defecto con el comando `userdel -r ubuntu`, con la opción `-r` para que borre su `home`. Es muy importante que antes de hacer esto nos aseguremos que el usuario `administrator` puede hacer sudo. En otro caso, al darse que `root` no tiene contraseña, perderíamos todo modo de hacer `sudo`, pues no podemos cambiar de usuario desde `administrator` a `root` ni logearnos directamente a `root`

El último detalle es editar el archivo `/etc/passwd` para cambiar el shell de nuestro usuario al que prefiramos, en nuestro caso, `bash`. Por defecto usa `sh`

Aspectos de Redes en la instalación del servidor

Protección básica del servidor y SSH

En lo que sigue, vamos a dar conexión a nuestro servidor para que sea alcanzable desde fuera de la red local. Por ello, hay que dar una seguridad básica para evitar ciertos ataques que comprometan nuestro servidor. Ahora mismo, el mayor problema que tenemos es que se pueden hacer ataques de fuerza bruta en `ssh` para logearse, probando todas las posibles contraseñas. Y esto es aún más peligroso teniendo en cuenta de que se puede hacer directamente sobre el usuario `root`. Aunque ahora esto último no se puede hacer, porque `root` no tiene contraseña, si en algún momento se le asigna una contraseña, esto sería posible. Por tanto, estos son los problemas más básicos (que ni remotamente son los únicos) que vamos a tratar de solventar.

Para ello vamos a usar `shh`. `ssh` puede usar claves de cifrado asimétrico. Para generar estas claves, desde nuestros ordenadores con los que accedemos al servidor, lanzamos el comando `ssh-keygen`, que genera los archivos `~/.ssh/id_rsa.pub` (clave pública) y `~/.ssh/id_rsa` (clave privada). La encriptación que se usa por defecto es RSA, aunque esto se puede cambiar. También se puede elegir una contraseña *passphrase*, esto hace que para poder leer la clave privada haya que introducir la contraseña, añadiendo un nivel extra de seguridad. Por tanto, si alguien consigue nuestra clave privada, no puede conectarse al servidor, porque conoce la clave que desencripta la propia clave privada. Tampoco la podría usar con otros fines.

Para que nuestro servidor nos acepte la clave `ssh`, usamos el comando `scp id_rsa.pub administrator@192.168.1.8:~/.ssh/sergio.pub` para copiar la clave pública. Una vez dentro del servidor, ejecutamos `cat sergio.pub >> authorized_keys` y ya podemos borrar el archivo copiado. A partir de este momento, el servidor ya no nos pedirá la clave al conectarnos desde nuestro ordenador al servidor a través del usurio `administrator`.

Esto no quita que se sigua pudiendo conectar usando una contraseña normal. Para evitar esto, editamos el archivo `/etc/ssh/sshd_config`. Para evitar que nadie, ya sea con contraseña normal o con clave `ssh`, se pueda conectar directamente al `root`, establecemos el siguiente campo: `PermitRootLogin: no`. Para forzar el uso de claves RSA como único método, establecemos el siguiente campo `PasswordAuthentification: no`

Con esto ya tenemos

Instalación de Nextcloud

Uso básico de Nextcloud

Análisis del sistema

Referencias