Laboratorio de administración y gestión de redes y sistemas
Examen de teoría
20 de enero de 2023.

Grado en Ingeniería Telemática, Universidad Rey Juan Carlos

- Entra en tu cuenta del laboratorio y ejecuta

  `~mortuno/prepara`

  Comprueba que esto deja en tu cuenta el fichero `~/lagrs.enero.23/teoria.TULOGIN.txt` (donde TULOGIN es tu nombre de usuario).

- Contesta ahí las cuatro preguntas siguientes:

Ejercicio 1. (2.5 puntos)

1.1) ¿Qué sucede cuando un usuario de Unix, distinto de *root*, ejecuta `su pedro`?

1.2) ¿Qué sucede cuando el usario *root* ejecuta `su pedro`?

1.3) ¿Qué sucede cuando un usuario de Unix, distinto de *root*, ejecuta `sudo su`?

1.4) ¿Qué sucede cuando el usario *root* ejecuta `sudo su`?

Respuesta:

1.1) El terminal le pide una contraseña, la del usuario *pedro*. Si se introduce correctamente, la sesión pasará a ser del usuario *pedro*.

1.2) La sesión pasará a ser del usuario *pedro*. El *root* puede, técnicamente, hacer cualquier cosa en el sistema, no necesita escribir ninguna contraseña.

1.3) Si el usuario está autorizado a ejecutar sudo, y escribe correctamente su propia contraseña, la sesión pasará a ser de *root*.

A menos que ya haya ejecutado con éxito alguna orden sudo recientemente (por omisión, 15 minutos). En este caso, pasará a ser *root* sin tener que escribirla contraseña de nuevo.

Para que un usuario pueda ejecutar *sudo*, debe estar dado de alta en el fichero `/etc/sudoers`. El primer usuario del sistema, el que ha instalado la máquina, siempre aparece ahí por omisión. En el caso de Ubuntu, el fichero *sudoers* está configurado para que, además, cualquier usuario que pertenezca al grupo *admin*, pueda ejecutar *sudo*. Aunque el grupo *admin* no está creado inicialmente.

1.4) Nada relevante, la sesión sigue siendo de *root*.

Ejercicio 2. (2.5 puntos)

Explica qué es, que hace, para qué sirve y dónde suele ejecutarse lo siguiente:
`source ~/.bashrc`

Respuesta:

Es una orden de shell Bash que ejecuta el script `.bashrc` del directorio *home* del usuario actual, pero dentro del entorno del proceso actual.

Sirve para que las funciones y las variables definidas en el fichero `.bashrc` sean visibles dentro de la shell actual. Si ejecutásemos un script desde otro script, lanzándolo escribiendo su nombre sin más, las fuciones, variables, alias, etc del segundo no serían accesibles en el primero.

Lo ejecuta automáticamente un sistema Unix/Linux con Bash en las shell interactivas no de login. Es conveniente que también se ejecute en las shell interactivas de login, por ejemplo las sesiones de ssh. Para ello es necesario que cada usuario añada esta línea a su fichero `.bash_profile`.

Ejercicio 3. (2.5 puntos)

Explica brevemente qué es el *desarrollo de software en cascada* y cuál es su relación con las técnicas *DevOps*.

Respuesta:

Es la técnica de desarrollo de software más habitual en los años 1980. De manera análoga a las ingenierías tradicionales, sigue de forma rígida y secuencial las fases de análisis de requerimientos, diseño, desarrollo, prueba, despligue y mantenimiento.

En el caso del software este enfoque resulta muy problemático, y como respuesta a sus inconvenientes, en los años 1990 surgen unas técnicas muy distintas denominadas de forma global *ágiles*. Estas técnicas de desarrollo de software tienen una filosofía que encaja bastante bien con las técnicas *DevOps*: flexibilidad, adaptación al cambio, orientación al código funcionando y colaboración entre las partes.

Ejercicio 4. (2.5 puntos)

4.1) ¿Para qué sirven los túneles ssh locales?

4.2) ¿Para qué sirven los túneles ssh remotos? (también llamados túneles inversos)

Respuesta

Ambos sirven para establecer tráfico seguro entre un cliente y un servidor, de forma similar a a una VPN aunque con una configuración muy sencilla de un único comando. A diferencia de una VPN, están limitados a TCP y solo un puerto.

- El túnel local asegura el tráfico entre el cliente y un proxy. El túnel remoto, entre un proxy y el servidor.
- El uso más típico de los túneles locales es proteger a un cliente que accede a internet desde un lugar especialmente inseguro, por ejemplo un *cibercafé* o un país sin garantías.
- El uso más típico de un túnel inverso es permitir ubicar un servidor TCP detrás de un NAT, cuando no podemos hacer *port forwarding (abrir puertos)* porque no somos los administradores de la red local. También es una forma sencila de distribuir la carga entre varios servidores, y/o añadir una capa de seguridad adicional.