**Administra directorios y archivos**

Antes, exploraste cómo administrar el sistema de archivos utilizando comandos de Linux. Conociste los siguientes comandos: **mkdir**, **rmdir**, **touch**, **rm**, **mv** y **cp**. En esta lectura, revisarás estos comandos, así como el editor de texto nano, y aprenderás otra forma de escribir en archivos.

**Cómo crear y modificar directorios**

**mkdir**

El comando **mkdir** crea un directorio nuevo. Al igual que todos los comandos que presentamos en esta lectura, puedes ingresar el directorio nuevo como la ruta de archivo absoluta, que comienza desde la raíz, o como una ruta de archivo relativa, que comienza desde el directorio actual.

Por ejemplo, si quieres crear un directorio nuevo llamado **network** en tu directorio **/home/analyst/logs**, puedes ingresar **mkdir /home/analyst/logs/network** para crear este directorio nuevo. Si ya estás en el directorio **/home/analyst/logs**, también puedes crear este directorio nuevo al ingresar **mkdir network**.

**Consejo profesional**: Puedes usar el comando **ls** para confirmar que se agregó el directorio nuevo.

**rmdir**

El comando **rmdir** elimina o borra un directorio. Por ejemplo, al ingresar **rmdir****/home/analyst/logs/network**, se eliminaría este directorio vacío del sistema de archivos.

**Nota**: El comando **rmdir** no puede eliminar directorios que contienen archivos o subdirectorios. Por ejemplo, si ingresas **rmdir /home/analyst**, obtendrás un mensaje de error.

**Cómo crear y modificar archivos**

**touch y rm**

El comando **touch** crea un archivo nuevo. Este archivo no tendrá ningún contenido dentro. Si tu directorio actual es **/home/analyst/reports**, al ingresar **touch permissions.txt**, se creará un archivo nuevo en el subdirectorio **reports**, llamado **permissions.txt**.

El comando **rm** elimina o borra un archivo. Este comando debe usarse con cuidado, ya que no es fácil recuperar archivos eliminados con **rm**. Para eliminar el archivo de permisos que acabas de crear, ingresa **rm** **permissions.txt**.

**Consejo profesional:** Para verificar que **permissions.txt** se haya creado o eliminado correctamente, puedes ingresar **ls**.

**mv y cp**

Si estás trabajando con archivos, también puedes usar **mv** y **cp**. El comando **mv** mueve un archivo o directorio a una ubicación nueva, y el comando **cp** copia un archivo o directorio a una ubicación nueva. El primer argumento después de **mv** o **cp** es el archivo o directorio que quieres mover o copiar, mientras que el segundo es la ubicación en la que quieres moverlo o copiarlo.

Para mover **permissions.txt** al subdirectorio **logs**, ingresa **mv permissions.txt /home/analyst/logs**. Al mover un archivo, se elimina de su ubicación original. Sin embargo, al copiarlo, esto no sucede. Para copiar **permissions.txt** en el subdirectorio **logs** y mantenerlo en su ubicación original, ingresa **cp permissions.txt /home/analyst/logs**.

**Nota**: El comando **mv** también puede usarse para cambiar el nombre de un archivo. Para hacerlo, ingresa el nombre nuevo como el segundo argumento en lugar de la ubicación nueva. Por ejemplo, al ingresar **mv permissions.txt** **perm.txt**, se cambia el nombre del archivo **permissions.txt** a **perm.txt**.

**Editor de texto nano**

**Nano** es un editor de archivos de línea de comandos disponible por defecto en muchas distribuciones de Linux. Muchos/as principiantes lo encuentran fácil de usar, y es muy utilizado en ciberseguridad. Puedes realizar varias tareas básicas en nano, como crear archivos nuevos y modificar contenido de archivos.

Para abrir un archivo existente en nano desde el directorio que lo contiene, ingresa **nano** seguido del nombre del archivo. Por ejemplo, al escribir **nano permissions.txt** desde el directorio **/home/analyst/reports**, se abre una nueva ventana de edición de nano con el archivo **permissions.txt** abierto para editar. También puedes proporcionar la ruta de acceso absoluta al archivo si no estás en el directorio que lo contiene.

Para crear un archivo nuevo en nano, ingresa **nano** seguido de un nombre de archivo nuevo. Por ejemplo, al ingresar **nano authorized\_users.txt** desde el directorio **/home/analyst/reports**, se crea el archivo **authorized\_users.txt** dentro de ese directorio y se abre en una nueva ventana de edición de nano.

Dado que no hay una función de guardado automático en nano, es importante que guardes tu trabajo antes de salir. Para guardar un archivo en nano, usa el atajo de teclado **Ctrl + O**. Se te pedirá que confirmes el nombre del archivo antes de guardarlo. Para salir de nano, utiliza el atajo de teclado **Ctrl + X**.

**Nota**: Otros editores de texto de línea de comandos populares son Vim y Emacs.

**Redireccionamiento de salida estándar**

Hay otra forma de escribir en archivos. Anteriormente, aprendiste sobre la entrada estándar y la salida estándar. La **entrada estándar** es información recibida por el sistema operativo a través de la línea de comandos, mientras que la **salida estándar** es la información devuelta por el SO a través del shell.

También aprendiste acerca del comando **pipe**, o pleca. El comando ***pipe*** envía la salida estándar de un comando como entrada estándar a otro comando, para su posterior procesamiento. Para usarlo, se ingresa el carácter pleca (**|**).

Además de la pleca (**|**), también puedes usar los operadores del signo “mayor que” (**>**) y el “doble mayor que” (**>>**) para redirigir la salida estándar.

Al usarlos con **echo**, los operadores **>** y **> >** pueden servir para enviar la salida de **echo** a un archivo específico en lugar de a la pantalla. La diferencia entre ambos es que **>** sobrescribe tu archivo existente, mientras que **>>** agrega tu contenido al final del archivo existente en lugar de sobrescribirlo. El operador **>** debe usarse con cuidado, ya que no es fácil recuperar archivos sobrescritos.

Cuando estés dentro del directorio que contiene el archivo **permissions.txt**, ingresa **echo "last updated** **date" >> permissions.txt** y agrega la cadena “last updated date” al contenido del archivo. Al ingresar **echo** **"time" > permissions.txt** después de este comando, se sobrescribe todo el contenido del archivo **permissions.txt** con la cadena “time”.

**Nota:** Los operadores **>** y **>>** crearán un archivo nuevo si todavía no existe uno con el nombre especificado.

**Conclusiones clave**

Saber cómo administrar el sistema de archivos en Linux es una habilidad importante para las/los analistas de seguridad. Estos son algunos comandos útiles para ello: **mkdir**, **rmdir**, **touch**, **rm**, **mv** y **cp**. Cuando un/a analista de seguridad necesita escribir en archivos, puede usar el editor de texto nano o los operadores **>** y **>>**.