



Powered By

# GNU/Linux



RedTauros Ltda

<http://www.redtauros.com>

## SERVIDOR SSH

OpenSSH es una completa colección de herramientas para controlar remotamente computadoras conectadas en red y transferir datos entre ellas.

OpenSSH es una versión gratuita y de código abierto de la familia de herramientas del protocolo Secure Shell (SSH). Herramientas tradicionales como telnet o rcp son inseguras y transmiten la contraseña del usuario en texto plano cuando se utilizan. OpenSSH proporciona un servicio (daemon) servidor y varias herramientas cliente para facilitar operaciones seguras y cifradas de control remoto y transferencia de archivos, reemplazando eficazmente a las herramientas antiguas.

El componente servidor de OpenSSH, sshd, está permanentemente escuchando peticiones de conexión de cualquier herramienta cliente. Cuando ocurre una solicitud de conexión, sshd establece la conexión adecuada según el tipo de cliente que se esté conectando. Por ejemplo, si la computadora remota utiliza la aplicación cliente SSH, el servidor OpenSSH configura una sesión de control remoto tras completar la autenticación. Si un usuario remoto se conecta a un servidor OpenSSH mediante scp, el demonio del servidor OpenSSH inicia una copia segura de archivos entre el servidor y el cliente tras la autenticación.

## Instalación

El objetivo del comando `sudo apt update` en Linux (específicamente en distribuciones basadas en Debian, como Ubuntu) es actualizar la lista de paquetes disponibles desde los repositorios configurados.

El sistema contacta a los servidores oficiales (y cualquier repositorio adicional que tengas configurado) y descarga la información más reciente sobre qué paquetes están disponibles y sus versiones, incluyendo actualizaciones de seguridad y nuevas versiones de software.

*`sudo apt update`*

El objetivo del comando `sudo apt upgrade` en Linux (en distribuciones basadas en Debian como Ubuntu) es actualizar los paquetes instalados en el sistema a sus últimas versiones disponibles.

El sistema revisa la lista de paquetes instalados y compara sus versiones con las disponibles en los repositorios actualizados (después de ejecutar `sudo apt update`). Si hay nuevas versiones (por ejemplo, por correcciones de errores, mejoras o actualizaciones de seguridad), te pregunta si deseas instalarlas.

*`sudo apt upgrade`*

Para instalar las aplicaciones cliente de OpenSSH en tu sistema Ubuntu, usa estos comandos en el terminal:

*`sudo apt install openssh-client`*

Para instalar la aplicación del servidor OpenSSH y los archivos de soporte relacionados, usa este comando en el terminal:

```
sudo apt install openssh-server
```

## Configurar OpenSSH

Para configurar el comportamiento predeterminado de la aplicación del servidor OpenSSH, sshd , edita el archivo `/etc/ssh/sshd_config` . Para obtener información sobre las directivas de configuración utilizadas en este archivo, consulta la página de manual en línea o ejecuta `man sshd_config` en una terminal.

Hay muchas directivas en el archivo de configuración de sshd , las cuales controlan aspectos como la configuración de comunicación y los modos de autenticación. A continuación, se muestran algunos ejemplos de directivas de configuración que puedes modificar editando el archivo `/etc/ssh/sshd_config` .

Identificar parámetros como:

1. Puerto
2. Direcciones que se atienden.
3. Banner

Antes de editar el archivo de configuración, deberías hacer una copia del archivo original `/etc/ssh/sshd_config` y protegerlo contra escritura para tener los ajustes originales como referencia y poder reutilizarlos si es necesario.

1. Identificar los permisos del archivo `sshd_config`
2. Crear una copia en la misma carpeta llamada `sshd_config.original`.
3. Quitar los permisos de escritura a `sshd_config.original` para que nadie lo pueda modificar.

Dado que perder acceso al servidor SSH podría significar perder la forma de conectarse a un servidor, verifica la configuración después de modificarla y antes de reiniciar el servicio:

```
sudo sshd -t -f /etc/ssh/sshd_config
```

modificar la siguiente linea de `sshd_config`

```
Include /etc/ssh/sshd_config.d/*.conf
```

Cambiarla a :

```
Cinclude /etc/ssh/sshd_config.d/*.conf
```

Volver a ejecutar :

```
sudo sshd -t -f /etc/ssh/sshd_config
```

Que pasó?

## Identificación de la IP

Para poder identificar la dirección IP del servidor, debemos de instalar el paquete *net-tools*

y ejecutar la instrucción:

```
ifconfig
```

## Conectandonos desde Windows

Usando PowerShell podemos usar el comando ssh que viene integrado, de la siguiente manera:

```
ssh usuario@ip
```

ó

```
ssh usuario@dominio
```

ó

```
ssh -p puerto usuario@dominio
```

nos pedirá el password, lo ingresamos y damos enter.

Estando en el Powershell de Windows hacemos lo siguiente:

Vamos a cambiar el mensaje de bienvenida del SSH, editando el archivo */etc/issue.net* adicionando las siguiente lineas:

```
Bienvenido al servidor Einstein de la universidad.
```

## Modificación directiva Banner:

Veamos un ejemplo de cambio en una directiva de configuración. Para que tu servidor OpenSSH muestre el contenido del archivo */etc/issue.net* como un mensaje antes del inicio de sesión (banner), puedes agregar o modificar la siguiente línea en el archivo */etc/ssh/sshd\_config*:

*Banner /etc/issue.net*

Después de realizar cambios en el archivo `/etc/ssh/sshd_config`, guarda el archivo. Reiniciar el servicio `sshd` para aplicar los cambios utilizando el siguiente comando:

*`sudo systemctl restart ssh.service`*

Salimos con exit del Powershell y volvemos a ingresar.

1. Identificar el mensaje de bienvenida.
2. Salir con exit

## Modificación directiva Port:

En el contexto de SSH (Secure Shell) , el puerto es el punto de comunicación a través del cual se establece la conexión segura entre un cliente y un servidor. Por defecto, SSH utiliza el puerto 22 , un número de puerto estandarizado que permite que los dispositivos identifiquen el servicio SSH dentro de una red.

Cambiar el puerto predeterminado de SSH es una práctica común para mejorar la seguridad del servidor , ya que ayuda a evitar ataques automatizados dirigidos al puerto 22. Aunque no brinda una protección completa, reduce la exposición a intentos de acceso no autorizados provenientes de bots o scripts maliciosos.

Para ello debemos de modificar el archivo de configuración de ssh:

*`/etc/ssh/sshd_config`*

Identificar la directiva Port, la cual esta comentada (#), quitarle el comentario y dejarlo en el puerto 1022, así:

*`Port 1022`*

Reiniciamos el servicio para que tenga efecto los cambios:

*`sudo systemctl restart ssh.service`*

Debemos de usar la opcion `-P`, a cual especifica el puerto a usar.

*`ssh -p puerto usuario@ip`*

# SCP

La instrucción `scp` (Secure Copy Protocol) es una herramienta utilizada en entornos Unix/Linux para copiar archivos y directorios de forma segura entre máquinas en una red, ya sea en una máquina local a una remota, o entre dos máquinas remotas. Este comando forma parte del conjunto de utilidades OpenSSH, lo que significa que utiliza el protocolo SSH (Secure Shell) para cifrar la transferencia de datos, garantizando así la seguridad y privacidad de la información transmitida.

A diferencia de comandos como `cp` (que solo opera en el sistema local) o protocolos inseguros como `ftp`, `scp` permite mover archivos entre sistemas de manera encriptada, protegiendo las credenciales y los datos del tráfico no seguro. Es especialmente útil cuando se trabaja en entornos de servidores remotos donde se requiere intercambiar archivos de manera rápida y segura.

Su sintaxis básica es similar a la siguiente:

```
scp origen usuario@destino:ruta
```

ó

```
scp usuario@destino:/ruta/documento .
```

( . ) significa, aquí, donde estoy ubicado.

Con `scp`, puedes copiar desde tu máquina local a un servidor remoto, desde un servidor remoto a tu máquina local, o incluso entre dos servidores remotos directamente desde tu terminal.

En esta guía exploraremos cómo usar `scp` con ejemplos prácticos, opciones útiles y buenas prácticas para aprovechar al máximo esta poderosa herramienta.

## Practicando SCP

Vamos a copiar este documento desde Windows a nuestro home en linux (/home/user/), de la siguiente manera:

1. Abrimos Powershell en Windows
2. Descargamos este PDF
3. Nos ubicamos en descargas
4. Ejecutamos:

```
scp .\Linux_serv_ssh.pdf usuario@IP:/home/usuario
```

5. Comprobamos en nuestro Linux la existencia del archivo.

## Practica

Realizar la practica en parejas.

1. Identificar la IP del compañero
2. Cambiar el puerto del ssh al 2022 y reiniciar el servicio.
3. Cambiar el mensaje de entrada diciendo : Este servidor esta monitoreado y cualquier intento de ataque tendrá todas las sanciones que la ley permita.
4. Pedirle que cree el usuario invitado con la clave yoamounicatolica.
5. Ingresar mediante ssh al equipo de su compañero.
6. Crear una carpeta llamada practica\_ssh
7. Dentro de practica\_ssh, crear dos archivos llamados sshd01 y sshd02
8. Adicionar dentro de sshd01 : SSH sirve para controlar remotamente computadoras conectadas en red y transferir datos entre ellas.
9. Adicionar dentro de sshd02 : para copiar archivos y directorios de forma segura entre máquinas en una red, ya sea en una máquina local a una remota, o entre dos máquinas remotas.
10. Agrupar la carpeta practica\_ssh en un archivo tar llamado practicassh\_01.tar ubicado en nuestro home (homer/user)
11. Ahora desde el cliente, vamos a descargar practicassh\_01.tar
12. Lo descomprimos y verificamos que tenga los contenidos.
13. Adicionamos un archivo llamado sshd03 a la carpeta practica\_ssh que acabamos de descomprimir.
14. Adicionar dentro de sshd03 : A diferencia de comandos como *cp* (que solo opera en el sistema local) o protocolos inseguros como *ftp*, *scp* permite mover archivos entre sistemas de manera encriptada.
15. Agrupar la carpeta practica\_ssh en un archivo tar llamado practicassh\_02.tar ubicado en nuestro home (homer/user)
16. Ahora desde el servidor, vamos a descargar practicassh\_02.tar
17. Lo descomprimos y verificamos que tenga los contenidos.

## Tips

La mejor forma de saber la dirección ip de un equipo en Linux, sin instalar paquete alguno, es usar el comando:

```
ip addr show
```

Otra forma de reiniciar los servicios seria con :

```
sudo systemctl stop ssh.service  
sudo systemctl start ssh.service
```

ó

```
service sshd restart
```

ó

```
service sshd stop  
service sshd start
```

## Acceso a una distribución de WSL 2 desde la red de área local (LAN)

Al usar una distribución de WSL 1, si su equipo estaba configurado para ser accesible desde su LAN, entonces las aplicaciones que se ejecutan en WSL también podrían ser accedidas en su LAN.

Este no es el caso predeterminado en WSL 2. WSL 2 tiene un adaptador ethernet virtualizado con su propia dirección IP única. Actualmente, para habilitar este flujo de trabajo, deberá seguir los mismos pasos que lo haría para una máquina virtual normal. (Estamos buscando maneras de mejorar esta experiencia).

A continuación se muestra un ejemplo de uso del comando Netsh interface portproxy Windows para añadir un proxy de puerto que escucha en el puerto de su host y conecta ese proxy de puerto a la dirección IP para la VM WSL 2.

*PowerShell*

```
netsh interface portproxy add v4tov4 listenport=<yourPortToForward> listenaddress=0.0.0.0  
connectport=<yourPortToConnectToInWSL> connectaddress=(wsl hostname -I)
```

En este ejemplo, deberá actualizar <yourPortToForward> a un número de puerto, por ejemplo, listenport=4000. listenaddress=0.0.0.0 significa que las solicitudes entrantes se aceptarán desde cualquier dirección IP. La dirección de escucha especifica la dirección IPv4 para la que se va a escuchar y se puede cambiar a valores que incluyen: dirección IP, nombre NetBIOS del equipo o nombre DNS del equipo. Si no se especifica una dirección, el valor predeterminado es el equipo local. Debe actualizar el valor de <yourPortToConnectToInWSL> a un número de puerto en el que desea que WSL se conecte, por ejemplo, connectport=4000. Por último, el connectaddress valor debe ser la dirección IP de la distribución de Linux instalada a través de WSL 2 (la dirección de máquina virtual WSL 2), que se puede encontrar escribiendo el comando: wsl.exe hostname -I.

Por lo tanto, este comando puede tener un aspecto similar al siguiente:

*PowerShell*

```
netsh interface portproxy add v4tov4 listenport=4000 listenaddress=0.0.0.0 connectport=4000  
connectaddress=192.168.101.100
```

Para obtener la dirección IP del WSL, use:

*PowerShell*

```
wsl hostname -I
```

```
cat /etc/resolv.conf para la dirección IP de la máquina Windows
```

Con listenaddress=0.0.0.0 se escuchará en todos los puertos IPv4.

Ejemplo:

*PowerShell*

```
netsh interface portproxy add v4tov4 listenport=1022 listenaddress=0.0.0.0 connectport=1022  
connectaddress=IP_WSL
```

Importante, bajar el firewall de windows.

Fuentes:

<https://documentation.ubuntu.com/server/how-to/security/openssh-server/index.html>

<https://learn.microsoft.com/es-es/windows/wsl/networking>

<https://www.youtube.com/watch?v=3Tkkl3ePZKo>