

PROTOCOLO SSH

SSH (Secure Shell) es un protocolo de red destinado a la conexión con máquinas a las que accedemos por línea de comandos, muy usado para administrar servidores Linux.

Su característica más importante es que siempre se realiza de manera segura. Ya que toda la información viaja encriptada, para evitar que nadie pueda escuchar el canal de comunicaciones para robar información o claves de acceso.

El puerto predeterminado para las conexiones SSH es el 22.

Funcionamiento:

El protocolo SSH utiliza una arquitectura cliente-servidor para establecer conexiones seguras. Aquí hay un resumen de cómo funciona:

- **Cliente SSH:** Es la aplicación que utilizas para conectarte a un servidor remoto. Puedes utilizar diferentes clientes SSH, como OpenSSH en sistemas Linux o PuTTY en Windows.
- **Servidor SSH:** Se ejecuta en el servidor remoto al que deseas acceder. Este servidor está configurado para aceptar conexiones SSH y autenticar a los usuarios.
- **Autenticación:** Cuando intentas conectarte a un servidor remoto, el cliente SSH y el servidor SSH inician un proceso de autenticación. Esto generalmente implica el uso de un nombre de usuario y una contraseña (o una clave SSH). La clave SSH es una forma más segura de autenticación y se recomienda encarecidamente su uso.

¿Para qué se utiliza SSH?

Debido a su seguridad, SSH es el modo preferido para la realización de conexión con servidores que necesitamos administrar. La diferencia con respecto a otros protocolos más antiguos como Telnet es que el protocolo SSH siempre es seguro.

Sin embargo, aprovechando la seguridad de las comunicaciones, también se utiliza para otros objetivos como:

- **Transferencia de Archivos Segura:** Permite transferir archivos de forma segura entre sistemas locales y remotos utilizando herramientas como el comando SCP o SFTP.
- **Creación de Túneles de Red:** SSH se utiliza para crear túneles de datos seguros que redirigen el tráfico de red a través de conexiones SSH, lo que puede ayudar a proteger la comunicación en redes no seguras. Se usan en sistemas como Ngrok, un software que permite a los desarrolladores exponer de manera remota los trabajos, tal como los tienen funcionando en su servidor de desarrollo local.

Técnicas de cifrado SSH

El protocolo SSH utiliza diferentes técnicas de seguridad para proteger sus conexiones.

Cifrado Simétrico

El cifrado simétrico es una técnica en la que se utiliza la misma clave tanto para cifrar como para descifrar los datos entre el cliente y el servidor, lo que garantiza su seguridad y confidencialidad.

Cifrado Asimétrico

En cambio, el cifrado asimétrico utiliza dos claves: una pública y otra privada; en otras palabras, se hace uso de una clave para el cifrado y otra para el descifrado, verificando así la identidad tanto del cliente como la del servidor.

Cuando un cliente se conecta a un servidor, utiliza la clave pública del servidor para cifrar un mensaje que sólo puede descifrarse con la clave privada correspondiente.

Hashing

El hashing es una técnica que se utiliza para verificar la integridad de los datos transmitidos. El algoritmo toma un conjunto de datos y genera un valor hash único, que es una representación de los datos originales.

Este valor se envía junto con los datos a través de la conexión SSH. En el extremo receptor, los datos se vuelven a calcular y se genera un nuevo valor hash.

- Si coincide con el recibido, se confirma que los datos no se han modificado.
- Si no coincide, los datos podrían haber sido alterados y se considera una posible amenaza de seguridad.

¿Cómo conectarse por SSH con servidores remotos?

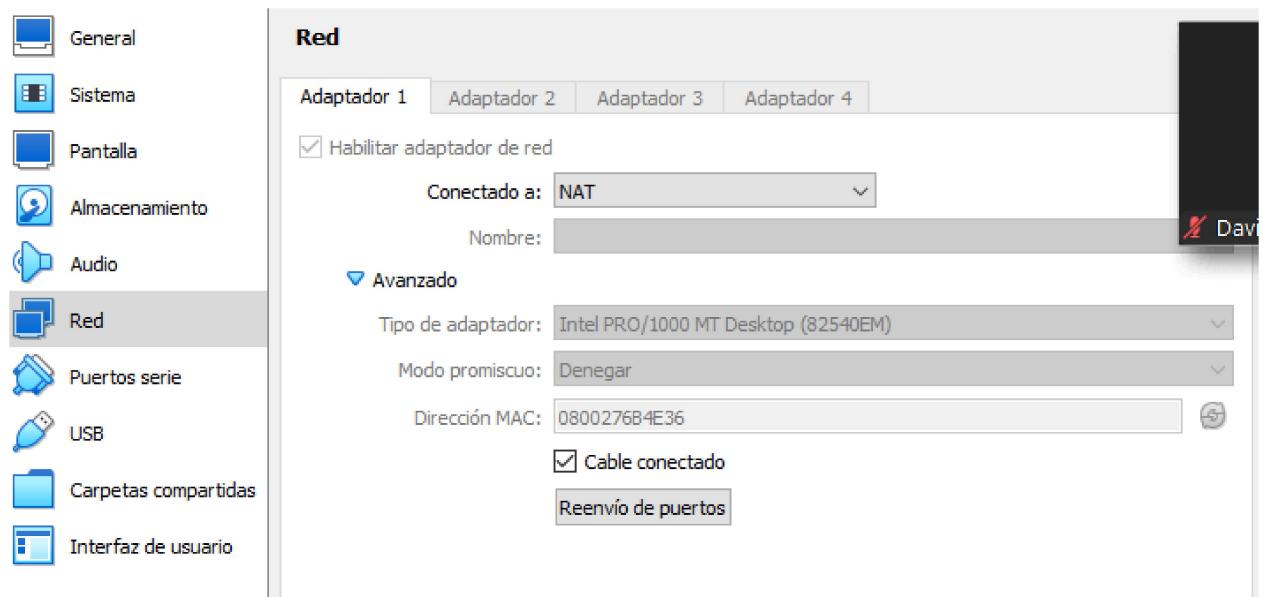
El comando para realizar la conexión para la administración remota de servidores se llama igual que el mismo protocolo: SSH. Para usar este comando necesitamos indicar tanto el usuario como la dirección IP o el nombre del host del servidor al que deseamos conectarnos.

```
C:\Users\hecto>ssh -p 2345 hectorserv@
```

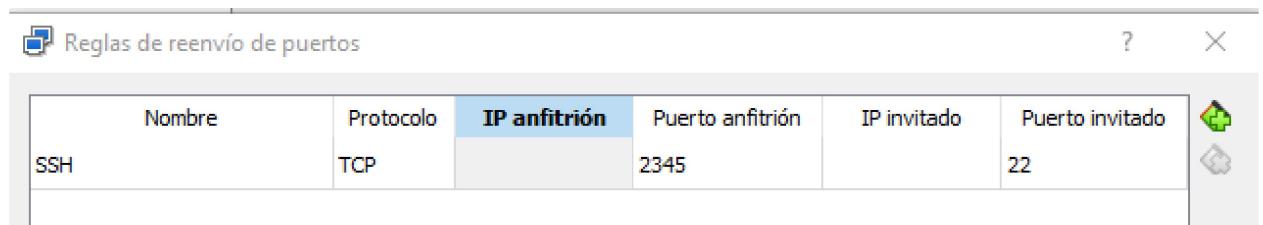
En el caso de una máquina en el mismo equipo es suficiente con indicar el usuario, con `-p` indicamos el puerto desde donde accedemos. Para ello vamos a ver como lo hemos configurado.

Configuración para máquina virtual

Por cifrado simétrico:



Accedemos a la configuración, vamos a la pestaña de red, pinchamos en avanzado y luego en Reenvío de puertos.



En la parte derecha pinchamos en agregar nueva regla, le damos un nombre, indicamos los puertos del anfitrión e invitado. Puerto anfitrión nuestro equipo, Puerto invitado, por cuál escucha la máquina, el 22 es por defecto.

Tras esto pasamos a configurar la máquina para que permita acceso por SSH

Configuración server Linux:

Actualizamos repositorios e instalamos OpenSSH, en un server suele venir instalado por defecto:

```
Leyendo lista de paquetes... Hecho  
hectorserv@hectorseerv:~$ sudo apt-get update
```

```
Leyendo lista de paquetes... Hecho  
hectorserv@hectorseerv:~$ sudo apt-get install openssh-server -y
```

Ahora debemos modificar el archivo de configuración para que escuche por el puerto 22, y permita conectarse con Login:

```
hectorserv@hectorseerv:~$ sudo nano /etc/ssh/sshd_config
```

Buscamos las líneas Port, ListenAddress y PasswordAuthentication, y las descomentamos. Con esto le indicamos que escuche por el puerto 22, a cualquier IP, y que permita logearse.

```
Port 22  
#AddressFamily any  
ListenAddress 0.0.0.0  
#ListenAddress ::
```

```
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication yes  
#PermitEmptyPasswords no
```

Guardamos y salimos.

Ahora hay que reiniciar el servicio para que actualice los cambios y comprobamos los cambios:

```
hectorserv@hectorseerv:~$ hectorserv@hectorseerv:~$ sudo systemctl restart ssh  
hectorserv@hectorseerv:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)  
  Active: active (running) since Mon 2024-10-14 10:00:16 UTC; 15s ago  
    Docs: man:sshd(8)  
          man:sshd_config(5)  
  Process: 2378 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
 Main PID: 2380 (sshd)  
    Tasks: 1 (limit: 4564)  
   Memory: 1.7M  
      CPU: 37ms  
     CGroup: /system.slice/ssh.service  
             └─2380 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
oct 14 10:00:16 hectorseerv systemd[1]: Stopping OpenBSD Secure Shell server...  
oct 14 10:00:16 hectorseerv systemd[1]: ssh.service: Deactivated successfully  
oct 14 10:00:16 hectorseerv systemd[1]: Stopped OpenBSD Secure Shell server.  
oct 14 10:00:16 hectorseerv systemd[1]: Starting OpenBSD Secure Shell server...  
oct 14 10:00:16 hektorseerv sshd[2380]: Server listening on 0.0.0.0 port 22.  
oct 14 10:00:16 hektorseerv systemd[1]: Started OpenBSD Secure Shell server.
```

Ya solo nos falta permitir la conexión en el Firewall:

```
hectorserv@hectorseerv:~$ sudo ufw allow ssh  
Skipping adding existing rule  
Skipping adding existing rule (v6)  
hectorserv@hectorseerv:~$ sudo systemctl restart ssh
```

Reiniciamos ssh para asegurnos.

Conexión SSH:

Ahora para conectarnos, accedemos a la terminal de nuestro equipo, e introducimos el siguiente comando:

```
ssh -p <puerto anfitrión> <usuario maquina>@<ip máquina**>
```

*** no es necesario si esta en nuestro equipo, además tendríamos que configurar la tarjeta de red de la máquina*

La primera vez que accedemos nos pide que confirmemos (yes), tras esto introducimos la password del usuario al que nos conectamos y voila.

```
C:\Users\hecto>ssh -p 2345 hectorserv@  
The authenticity of host '[ ]:2345 ([192.168.1.49]:2345)' can't be established.  
ED25519 key fingerprint is SHA256:JwdS+JY1lV1+wV8fxvTXFya5D7ooERQxA4ls4f0pwYY.  
This host key is known by the following other names/addresses:  
    C:\Users\hecto/.ssh/known_hosts:7: [ ]:2222  
    C:\Users\hecto/.ssh/known_hosts:10: [192.168.1.49]:2222  
    C:\Users\hecto/.ssh/known_hosts:11: [192.168.1.49]:2345  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '[ ]:2345' (ED25519) to the list of known hosts.  
hectorserv@'s password:  
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-122-generic x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:     https://landscape.canonical.com  
 * Support:        https://ubuntu.com/pro  
  
System information as of lun 14 oct 2024 09:39:17 UTC  
  
 System load:  0.08          Processes:           109  
 Usage of /:   39.0% of 19.14GB  Users logged in:      0  
 Memory usage: 5%            IPv4 address for enp0s3: 10.0.2.15  
 Swap usage:   0%  
  
El mantenimiento de seguridad expandido para Applications está desactivado  
  
Se pueden aplicar 5 actualizaciones de forma inmediata.  
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable  
  
Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.  
Vea https://ubuntu.com/esm o ejecute «sudo pro status»  
  
New release '24.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Mon Oct 14 09:31:42 2024 from 10.0.2.2  
hectorserv@hectorserv:~$
```

Activar Windo
Ve a Configuración