

Comunicación SSH Linux-Windows

Paso 1. Primero nos aseguramos que los equipos que se comunicaran por ssh estén en la misma red, tanto el equipo Windows como el de Linux.

Ip para el equipo Debian

```
amf@debian: ~  
root@debian:~# ip ad  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 08:00:27:70:9c:56 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.137.15/24 brd 192.168.137.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 604021sec preferred_lft 604021sec  
    inet6 fe80::7f32:df7a:5f6f:4dc2/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
root@debian:~#
```

La ip asignada por DHCP en el equipo Windows

```
Adaptador de LAN inalámbrica Wi-Fi:  
  
Sufijo DNS específico para la conexión. . : mshome.net  
Vínculo: dirección IPv6 local. . . : fe80::24d:e05a:3b21:8b05%13  
Dirección IPv4. . . . . : 192.168.137.57  
Máscara de subred . . . . . : 255.255.255.0  
Puerta de enlace predeterminada . . . . . : 192.168.137.1
```

Como vemos, el segmento de red es el mismo para los dos equipos.

Paso 2. Partiendo por linux primeramente, instalamos un servidor SSH en el equipo Debian, instalamos OpenSSH mediante el siguiente comando:

```
root@debian:~# sudo apt-get install openssh-server
```

Una vez instalado el servidor, nos aseguramos que el firewall permita conexiones entrantes en el puerto 22, que es el puerto por defecto para SSH. Podemos permitir el acceso en el firewall de Debian mediante el siguiente comando en la terminal de Debian:

```
root@debian:~# sudo ufw allow 22/tcp  
Skipping adding existing rule  
Skipping adding existing rule (v6)  
root@debian:~#
```

Y comprobamos los puertos habilitados

```

root@debian:~# ufw status
Status: active

To Action From
--
80/tcp ALLOW Anywhere
443/tcp ALLOW Anywhere
WWW Full ALLOW Anywhere
22/tcp ALLOW Anywhere
80/tcp (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)
WWW Full (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)

```

En el equipo con Windows, podemos descargar un cliente SSH. Uno de los clientes más populares es PuTTY, que puedes descargar desde su sitio web oficial. Pero lo podemos conectar directamente desde el cmd de Windows.

Para verificar que hay comunicación entre el cliente y el servidor SSH, realizamos la prueba con el comando ping y vemos que los paquetes enviados son recepcionados correctamente.

Primero comprobamos del cliente de Windows al Debian

```

C:\Users\MARIA>ping 192.168.137.15

Haciendo ping a 192.168.137.15 con 32 bytes de datos:
Respuesta desde 192.168.137.15: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.137.15: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.137.15: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.137.15: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 192.168.137.15:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
        (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms

```

Vemos que hay 4 paquetes recibidos por parte del equipo Debian, ahora verificamos al equipo cliente Windows.

```
amf@debian: ~  
root@debian:~# ping -c 4 192.168.137.57  
PING 192.168.137.57 (192.168.137.57) 56(84) bytes of data.  
64 bytes from 192.168.137.57: icmp_seq=1 ttl=128 time=0.381 ms  
64 bytes from 192.168.137.57: icmp_seq=2 ttl=128 time=1.03 ms  
64 bytes from 192.168.137.57: icmp_seq=3 ttl=128 time=0.931 ms  
64 bytes from 192.168.137.57: icmp_seq=4 ttl=128 time=0.889 ms  
  
--- 192.168.137.57 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3329ms  
rtt min/avg/max/mdev = 0.381/0.808/1.031/0.251 ms  
root@debian:~#
```

Igualmente vemos que ningún paquete se perdió.

Ahora probaremos la conexión SSH desde el cliente de Windows.

```
C:\Users\MARIA>ssh root@192.168.137.15 -p 22  
The authenticity of host '192.168.137.15 (192.168.137.15)' can't be established.  
ECDSA key fingerprint is SHA256:D5bDPXpX28ZRZX3ehWSc6/fNigzx2AbfI5v21si0J78.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.137.15' (ECDSA) to the list of known hosts.  
root@192.168.137.15's password:
```

Vemos que se establece la conexión ingresando con el usuario root, ahora nos pedirá la contraseña que establecimos en Debian con el usuario root, igualmente podemos ingresar con otro usuario creado en nuestro equipo, pero hay que ver que dicho usuario tenga los privilegio y permisos respectivos para visualizar y modificar algún archivo.

```
OpenSSH SSH client  
C:\Users\MARIA>ssh root@192.168.137.15 -p 22  
root@192.168.137.15's password:  
Linux debian 5.10.0-21-amd64 #1 SMP Debian 5.10.162-1 (2023-01-21) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Apr 6 15:22:42 2023 from 192.168.16.52  
root@debian:~# who  
amf      tty2      2023-04-25 08:58 (tty2)  
root     pts/1      2023-04-25 12:37 (192.168.137.57)  
root@debian:~# ls /etc/network  
if-down.d if-post-down.d if-pre-up.d if-up.d interfaces interfaces.d interfaces.save  
root@debian:~#
```

Como comprobamos, pudimos establecer la conexión por SSH al servidor Debian y verificamos que las peticiones por comando que realizamos están correctas para el cliente conectado.