



Actividad #3: “Servicio SSH”

**Centro Universitario de Ciencias
Exactas e Ingenierías**

Departamento de Computación

Ingeniería en Informática

Padilla Manzo Juan Carlos

Código: 214610235

Materia: Adm. Servidores

Sección: D07

Fecha: 11/09/2021

¿Qué es SSH?

Secure Shell o en sus siglas SSH, es un protocolo que permite la administración remota, con esto los usuarios pueden controlar y modificar los servidores que tengan con solo tener conexión a internet. El protocolo SSH se inventó para tener una manera más segura de hacer una conexión remota, esto en sustitución Telnet, SSH a diferencia de Telnet, no cifra, pero sí utiliza técnicas criptográficas para asegurar que las comunicaciones que ocurran hacia o desde el servidor, estén siempre cifradas. Tiene seguridad para que no cualquier persona pueda ingresar al servidor, tiene maneras de autenticar los usuarios que quieran acceder de manera remota, proporciona mecanismos para poder transferir entradas de un cliente hacia el host y retransmitir una salida de vuelta al cliente.

¿Cómo funciona?

SSH no es un programa, es un protocolo, por lo que no es necesario que instalemos ningún software para poder activarlo. Al activar el protocolo SSH estamos indicando que en el equipo que lo activamos vamos a hacer una conexión remota con el equipo de forma segura y con cifrado.

En caso de que estemos utilizando Max o Linux, no será necesario descargar ningún programa, lo podremos hacer desde la terminal de nuestros sistemas sin ningún problema, solo ejecutando un comando, en cambio, si queremos hacer el acceso desde Windows, tendremos que utilizar algún programa (cliente SSH) que nos permita abrir conexiones SSH. Un ejemplo de estos programas es PuTTY, es bastante sencillo de utilizar y nos dará la misma comodidad que podríamos tener en Linux o Mac.

Para iniciar, ejecutaremos “apt install openssh-server openssh-client”.

```
root@servidor:~# apt-get install openssh-server openssh-client
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssh-client ya está en su versión más reciente (1:7.9p1-10+deb10u2).
Paquetes sugeridos:
  molly-guard monkeysphere rssh ssh-askpass ufw
Se instalarán los siguientes paquetes NUEVOS:
  openssh-server openssh-sftp-server
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 70 no actualizados.
Se necesita descargar 397 kB de archivos.
Se utilizarán 1 609 kB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian oldstable/main amd64 openssh-sftp-server amd64
4 1:7.9p1-10+deb10u2 [44.6 kB]
Des:2 http://deb.debian.org/debian oldstable/main amd64 openssh-server amd64 1:7
.9p1-10+deb10u2 [352 kB]
Descargados 397 kB en 0s (1 075 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete openssh-sftp-server previamente no seleccionado.
(Leyendo la base de datos ... 133089 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../openssh-sftp-server_1%3a7.9p1-10+deb10u2_amd64
```

Editaremos el archivo “sshd_config”

```
...
root@servidor:~# nano /etc/ssh/sshd_config
```

Habilitamos el puerto 22 descomentando la línea “Port 22” y agregaremos las 2 últimas líneas con nuestra dirección IP del servidor.



```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 3.2 /etc/ssh/sshd config Modificado

# $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
ListenAddress 192.168.17.128
Protocol 2
```

Habilitamos la línea ...

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

Agregamos la línea allowUsers con el correspondiente nombre de usuario



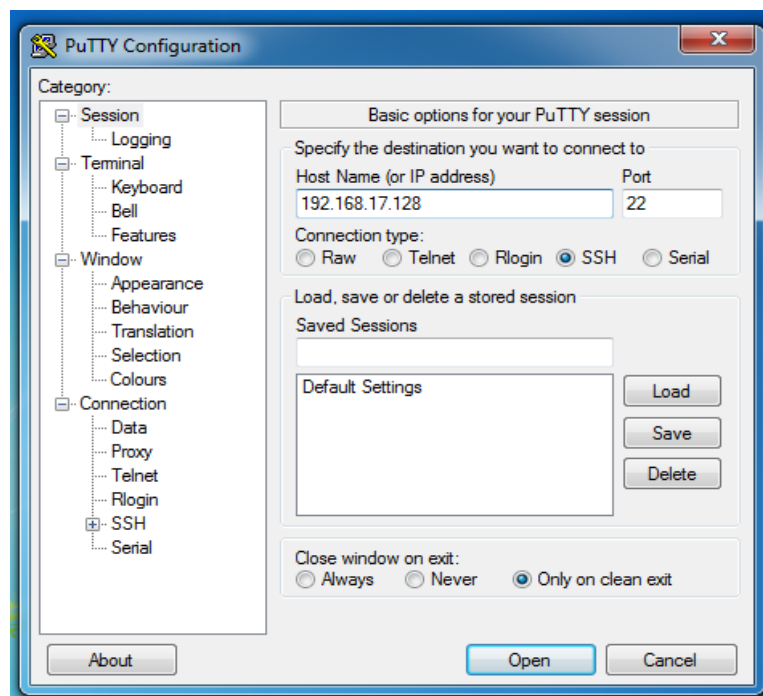
```
juan@servidor: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
GNU nano 3.2 /etc/ssh/sshd_config Modificado  
  
#ClientAliveCountMax 3  
#UseDNS no  
#PidFile /var/run/sshd.pid  
#MaxStartups 10:30:100  
#PermitTunnel no  
#ChrootDirectory none  
#VersionAddendum none  
  
AllowUsers juan
```

Ctrl + O y Ctrl + X para guardar y salir respectivamente.

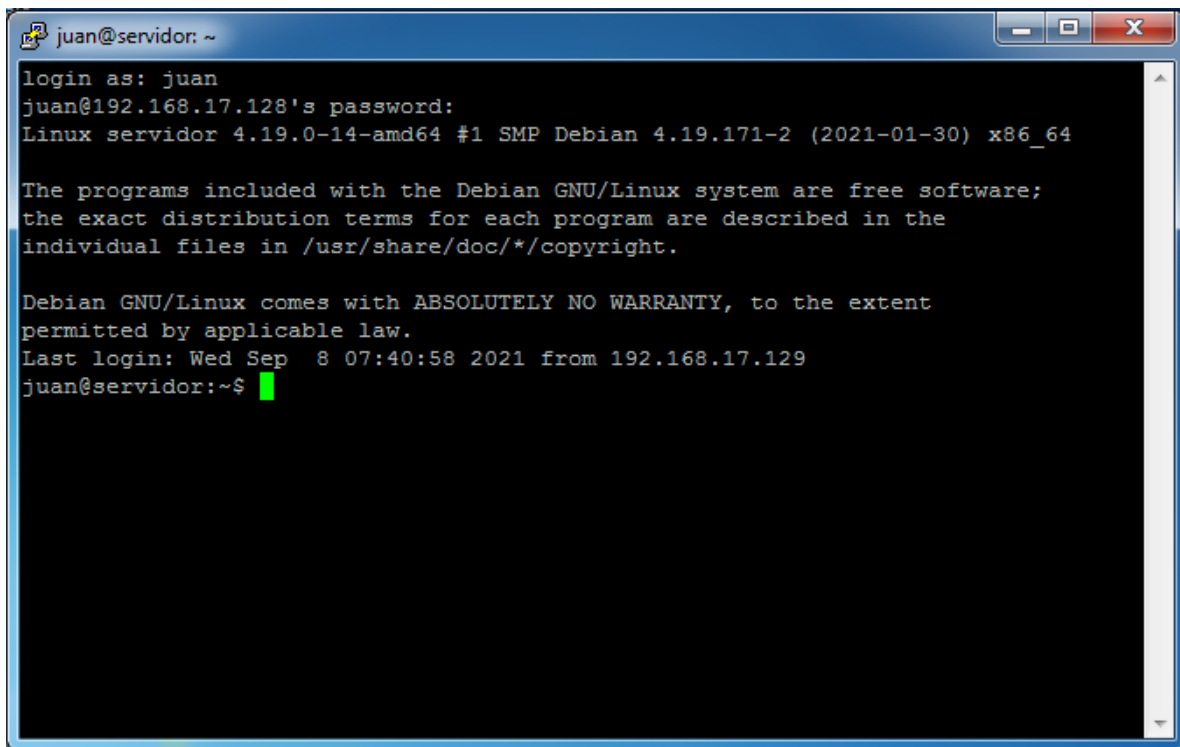
Y reiniciamos el servidore ssh con el comando:

```
root@servidor:~# service ssh restart
```

Accederemos a PuTTY desde Windows y ponemos la IP ...



Le damos en Open y se nos abrirá una ventana, en la cual tendremos que acceder con nuestro usuario y contraseña

A screenshot of a terminal window titled 'juan@servidor: ~'. The window shows the output of an SSH login session. The text displayed is: 'login as: juan', 'juan@192.168.17.128's password:', 'Linux servidor 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) 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: Wed Sep 8 07:40:58 2021 from 192.168.17.129', and 'juan@servidor:~\$' followed by a green cursor. The terminal has a blue title bar and standard window controls (minimize, maximize, close) in the top right corner.

```
login as: juan
juan@192.168.17.128's password:
Linux servidor 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) 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: Wed Sep 8 07:40:58 2021 from 192.168.17.129
juan@servidor:~$
```

CONCLUSIÓN

En esta actividad pudimos ver como el servicio de PuTTY es más práctico, y aunque la actividad fue sencilla, las ventajas que conllevan instalar el servicio SSH son bastantes, ya que podremos acceder a nuestro servidor de manera remota, aunque hay que decir que encontré unas curiosidades, y es que para Mac no es necesario tener el programa de PuTTY, ya que se puede hacer desde la terminal, lo que no fue tan sorprendente es que también es posible hacer esto desde Linux, lo cual resultaba un poco obvio sí se sabe que Linux se usa regularmente para el uso y desarrollo de los servidores.