

SSH + SSL EN SERVIDOR LINUX UBUNTU SERVER

(Rafa Polo)

¿Qué herramientas vamos a usar?

OPENSSL

Consiste en un robusto **paquete de herramientas** de administración y bibliotecas **relacionadas con la [criptografía](#)**, que suministran funciones criptográficas a otros paquetes como [OpenSSH](#) y navegadores web (para **acceso seguro a sitios [HTTPS](#)**).

Estas herramientas **ayudan al sistema a implementar** el *Secure Sockets Layer* ([SSL](#)), así como otros **protocolos relacionados con la seguridad**, como el *Transport Layer Security* (TLS). **OpenSSL también permite crear certificados digitales** que pueden aplicarse a un servidor, por ejemplo [Apache](#).

SSH

Las siglas corresponden a Secure SHell. Sirve para **acceder a máquinas remotas**, igual que hace telnet, pero de una **forma segura** ya que la **conexión va cifrada**. El transporte se hace **mediante TCP**, por tanto nos garantiza que las órdenes van a llegar a su destino (conectivo, fiable, orientado a conexión)

El cifrado de **SSH proporciona autenticidad e integridad de los datos** transmitidos por una red insegura como internet.

Utiliza llaves públicas para la autenticación en la máquina remota.

SSH no sólo sirve para usar comandos en máquinas remotas, sino para **transferencias de ficheros de forma segura ya sea por SCP o sFTP y servicios de escritorio remoto**.

1. Actualizamos paquetería y sistema

```
rafa@ubuntuserver:~$  
rafa@ubuntuserver:~$  
rafa@ubuntuserver:~$  
rafa@ubuntuserver:~$ sudo apt-get update && apt-get upgrade
```

2. Instalamos openssh-server

```
rafa@ubuntuserver:~$  
rafa@ubuntuserver:~$ sudo apt-get install openssh-server  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias
```

3. Instalamos openssl

```
rafa@ubuntuserver:~$ sudo apt-get install openssl  
[sudo] password for rafa:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias
```

4. Editamos la configuración del servidor ssh

sudo nano /etc/ssh/sshd_config

*Puerto por donde va a escuchar nuestro servidor y trae por defecto el puerto 22. IMPORTANTE cambiarlo (ejemplo puerto 8756) para mayor seguridad.

```
# What ports, IPs and protocols we listen for  
Port 22  
# The following lines define the port forwarding and X11 forwarding  
# capabilities. You should carefully consider whether you need them.  
# If you do not need them, you can comment them out.  
# ForwardAgent no  
# ForwardX11 no  
# ForwardX11Trusted yes  
# LocalForward no  
# RemoteForward no  
# What ports, IPs and protocols we listen for  
Port 8756  
# The following lines define the port forwarding and X11 forwarding  
# capabilities. You should carefully consider whether you need them.  
# If you do not need them, you can comment them out.  
# ForwardAgent no  
# ForwardX11 no  
# ForwardX11Trusted yes  
# LocalForward no  
# RemoteForward no
```

*Usaremos el **protocolo 2 de SSH**, mucho más seguro, por tanto forzaremos a que siempre conecten por protocolo 2.

```
#ListenAddress ::  
#ListenAddress 0.0.0.0  
Protocol 2  
# The following lines define the port forwarding and X11 forwarding  
# capabilities. You should carefully consider whether you need them.  
# If you do not need them, you can comment them out.  
# ForwardAgent no  
# ForwardX11 no  
# ForwardX11Trusted yes  
# LocalForward no  
# RemoteForward no
```

*Lugar donde se guardan las keys.

```
# HostKeys for protocol version 2  
HostKey /etc/ssh/ssh_host_rsa_key  
HostKey /etc/ssh/ssh_host_dsa_key  
HostKey /etc/ssh/ssh_host_ecdsa_key  
HostKey /etc/ssh/ssh_host_ed25519_key
```

*Privilege Separation activado por seguridad

```
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes
```

*Aquí le podemos indicar el tiempo que tiene un usuario para hacer login correctamente antes de que se le cierre automáticamente la sesión.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin without-password
StrictModes yes
```

*Aquí **decidimos si queremos** logarnos con **contraseña** root o sin contraseña:

- Without-password (sin contraseña)

-no (necesitaría contraseña)

```
# Authentication:
LoginGraceTime 120
PermitRootLogin without-password
StrictModes yes
```

5. Arranque y parada del servidor SSH

INICIAR O REINICIAR:

```
sudo /etc/init.d/ssh restart
```

PARAR EL SERVIDOR:

```
sudo /etc/init.d/ssh stop
```

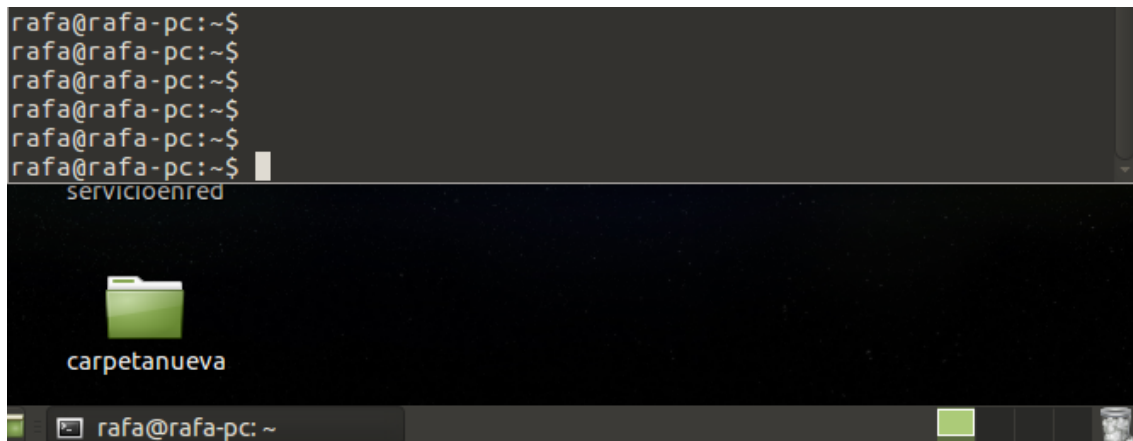
```
rafa@ubuntuuserver:~$ sudo /etc/init.d/ssh restart
[sudo] password for rafa:
ssh stop/waiting
ssh start/running, process 1246
rafa@ubuntuuserver:~$ sudo /etc/init.d/ssh stop
ssh stop/waiting
rafa@ubuntuuserver:~$ _
```

6. CONEXIÓN AL SERVIDOR mediante SSH

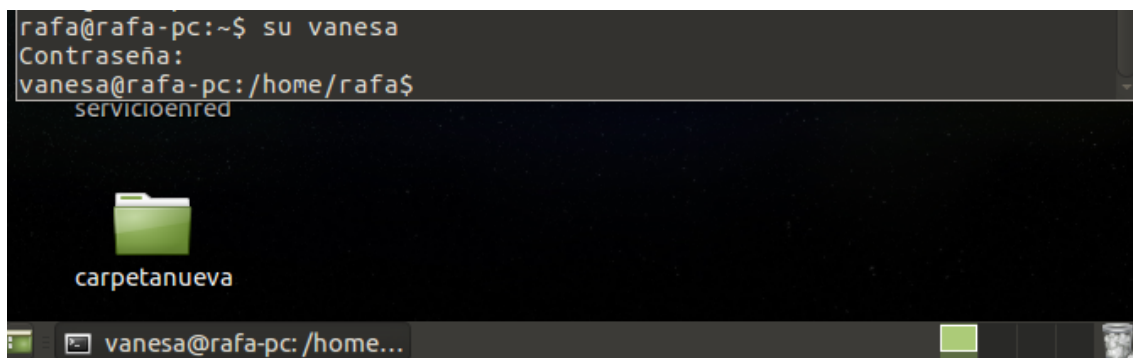
Para conectar desde un PC cliente al servidor mediante ssh, debemos ejecutar el comando ssh seguido del nombre o dirección IP del servidor. La conexión se realizará con el mismo nombre de usuario que estemos utilizando en el PC cliente.

EN CLIENTE, vemos cómo se da la conexión remota al servidor:

*Terminal de Ubuntu MATE con usuario administrador (rafa)



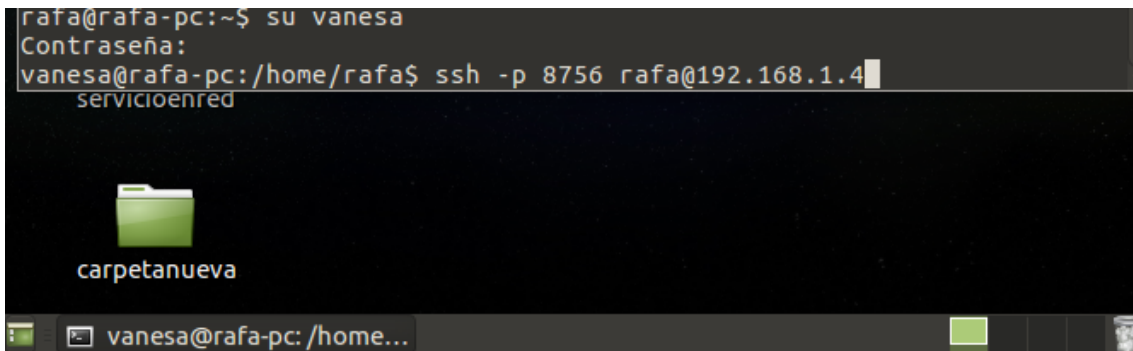
*Cambiamos de usuario (tenemos dos usuarios en nuestro Ubuntu Mate) para hacer la conexión desde usuario vanesa



*Accedemos al servidor con la siguiente línea de comando, donde:

- "ssh" es el comando es sí.
- "-p" indica a SSH que utilice un puerto no privilegiado (para conexiones tras router o firewall que bloquean conexiones a puertos privilegiados (<1024))
- "usuario" nombre de usuario en el servidor
- "ipservidor" como su nombre indica, la IP asignada al servidor y con la que se conecta a la red.

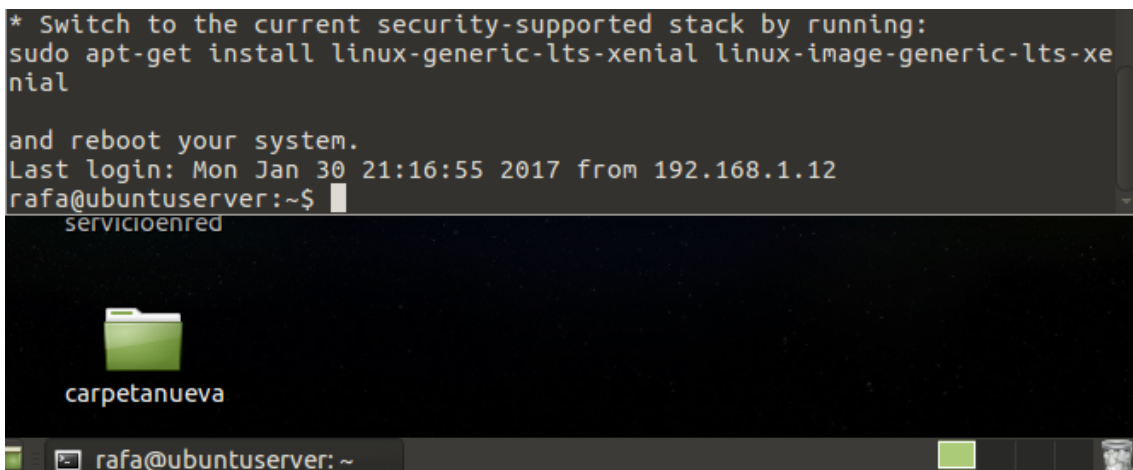
```
rafa@rafa-pc:~$ su vanesa
Contraseña:
vanesa@rafa-pc:/home/rafa$ ssh -p 8756 rafa@192.168.1.4
servicioenred
```



*Ya estamos en remoto


```
* Switch to the current security-supported stack by running:
sudo apt-get install linux-generic-lts-xenial linux-image-generic-lts-xe
nial

and reboot your system.
Last login: Mon Jan 30 21:16:55 2017 from 192.168.1.12
rafa@ubuntuserver:~$
servicioenred
```



**NOTA: si usamos por defecto el puerto 22 y no estamos tras un router o firewall que no permitan conexiones a puertos privilegiados, nos servirá de la siguiente forma:*


```
vanesa@rafa-pc:/home/rafa$ ssh rafa@192.168.1.4
The authenticity of host '192.168.1.4 (192.168.1.4)' can't be established.
ECDSA key fingerprint is SHA256:sKjKV2JBCmWqTDNpLe0Tz1ePPjoLRQCZ3SI2BSkcZDM.
Are you sure you want to continue connecting (yes/no)? yes
servicioenred
```



carpetanueva

De nuevo en remoto:

```
and reboot your system.
Last login: Mon Jan 30 21:31:15 2017 from 192.168.1.12
rafa@ubuntuserver:~$
servicioenred
```



carpetanueva